Web Intelligence et Science des Données (WISD)
Master International Francophone en double diplomation avec l'Université Sorbonne Paris Nord

Analyse de régression logistique pour prédire le risque de diabète : Étude de cas sur la population indienne PIMA

Réalisée par :

BENALI Ghita & **SGHIR Ghita**

Encadré par :

ALJ Abd

WISD 2023-2024

Remerciements

Nous exprimons notre profonde gratitude envers tous ceux qui ont collaboré de près ou de loin à la réalisation de ce projet. Nous sommes reconnaissants envers Allah le Tout-Puissant pour nous avoir accordé le courage et la détermination nécessaires à la réalisation de ce travail. Nous tenons à remercier sincèrement notre encadrant, M. ALJ Abdelkamel, pour sa supervision, sa disponibilité et ses précieux conseils scientifiques tout au long de cette étude. Enfin, nous exprimons notre profonde reconnaissance envers nos professeurs de la Faculté des Sciences de Dhar El-Mahraz pour nous avoir offert la possibilité d'acquérir une formation professionnelle.

Introduction générale

Le diabète sucré est devenu un problème de santé publique majeur dans le monde entier, avec des implications significatives pour la qualité de vie des individus et les systèmes de santé. La prédiction précoce du risque de diabète chez les individus peut jouer un rôle crucial dans la prévention et la gestion de cette maladie chronique. Dans cette étude, nous examinerons comment l'analyse de régression logistique peut aider à prédire le risque de diabète au sein de la population indienne.

Nous utiliserons l'ensemble de données Pima Indian Diabetes, qui contient des informations essentielles sur les patients indiens Pima, comme base pour notre analyse. Cette base de données nous permettra d'explorer les relations entre différentes variables et de comprendre les facteurs qui contribuent au risque de diabète au sein de cette population spécifique.

Le présent rapport vise à explorer l'application de la régression logistique pour prédire le risque de diabète chez la population indienne Pima , nous commencerons dans la première partie par explorer les données, en menant une analyse descriptive et en visualisant les relations entre les variables. Ensuite, nous construirons un modèle de régression logistique pour prédire le risque de diabète chez les patients. Nous évaluerons ensuite les performances de notre modèle en utilisant différentes mesures, telles que la précision, le rappel, etc.

Table des matières

Table des figures

1 Chapitre : Prétraitement des données

1.1 Introduction

Le prétraitement des données constitue une étape essentielle dans le domaine de l'analyse des données et de l'apprentissage automatique. Avant de pouvoir exploiter pleinement nos données, il est impératif de les préparer et de les nettoyer afin d'assurer leur qualité et leur cohérence. Ce chapitre se focalise sur les techniques de prétraitement des données qui améliorent la fiabilité et l'efficacité de nos modèles d'apprentissage automatique. Nous examinerons en détail les différentes phases du prétraitement, allant de la collecte et de la compréhension des données à la normalisation, à la transformation des caractéristiques, et à la sélection des variables les plus pertinentes. En appliquant ces méthodes, nous pourrons exploiter pleinement le potentiel de nos données pour obtenir des résultats précis et fiables.

1.2 Description du dataset

Le dataset utilisé dans cette étude est la base de données « Pima Indians Diabetes Database ». Cette ressource est largement étudiée et reconnue dans le domaine de la recherche sur le diabète. Initialement recueillie par le National Institute of Diabetes and Digestive and Kidney Diseases, cette base de données est souvent utilisée comme référence pour évaluer les algorithmes de prédiction du diabète.

La Pima Indians Diabetes Database contient des informations médicales sur un groupe de femmes indiennes Pima, parmi lesquelles certaines ont développé le diabète de type 2.

Notre dataset contient 9 variables, où "Outcome" est la variable réponse et les autres sont des variables explicatives :

- Pregnancies : Nombre de grossesses.
- Glucose : Concentration plasmatique de glucose après un test de tolérance au glucose oral.
- BloodPressure : Pression artérielle diastolique (mm Hg).
- SkinThickness : Épaisseur du pli cutané au niveau du triceps (mm).
- Insulin : Insuline sérique (mu U/ml).
- BMI : Indice de masse corporelle (poids en kg / (taille en m)^2).
- DiabetesPedigreeFunction : Fonction du pedigree du diabète, qui fournit une mesure de la susceptibilité génétique au diabète en fonction de l'historique familial.
- Age : Âge en années.
- Outcome : Variable cible binaire indiquant si le patient a développé le diabète (1) ou non

(0).

```
$ Pregnancies              : int  6 1 8 1 0 5 3 10 2 8 ...
$ Glucose                  : int  148 85 183 89 137 116 78 115 197 125 ...
$ BloodPressure            : int  72 66 64 66 40 74 50 0 70 96 ...
$ SkinThickness            : int  35 29 0 23 35 0 32 0 45 0 ...
$ Insulin                  : int  0 0 0 94 168 0 88 0 543 0 ...
$ BMI                      : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
$ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
$ Age                      : int  50 31 32 21 33 30 26 29 53 54 ...
$ Outcome                  : int  1 0 1 0 1 0 1 0 1 1 ...
```

FIGURE 1 – Description du dataset

1.3 Information sur la Dataset

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0
7	3	78	50	32	88	31.0	0.248	26	1
8	10	115	0	0	0	35.3	0.134	29	0
9	2	197	70	45	543	30.5	0.158	53	1
10	8	125	96	0	0	0.0	0.232	54	1
11	4	110	92	0	0	37.6	0.191	30	0
12	10	168	74	0	0	38.0	0.537	34	1
13	10	139	80	0	0	27.1	1.441	57	0
14	1	189	60	23	846	30.1	0.398	59	1
15	5	166	72	19	175	25.8	0.587	51	1
16	7	100	0	0	0	30.0	0.484	32	1
17	0	118	84	47	230	45.8	0.551	31	1

FIGURE 2 – Statistical Overview of the Dataset

1.4 Vue d'ensemble statistique du jeu de données

```
 Pregnancies        Glucose      BloodPressure    SkinThickness      Insulin          BMI
Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0   Min.   : 0.00
1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0   1st Qu.:27.30
Median : 3.000   Median :117.0   Median : 72.00   Median :23.00   Median : 30.5   Median :32.00
Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54   Mean   : 79.8   Mean   :31.99
3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2   3rd Qu.:36.60
Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0   Max.   :67.10
DiabetesPedigreeFunction      Age          Outcome
Min.   :0.0780           Min.   :21.00   Min.   :0.000
1st Qu.:0.2437           1st Qu.:24.00   1st Qu.:0.000
Median :0.3725           Median :29.00   Median :0.000
Mean   :0.4719           Mean   :33.24   Mean   :0.349
3rd Qu.:0.6262           3rd Qu.:41.00   3rd Qu.:1.000
Max.   :2.4200           Max.   :81.00   Max.   :1.000
```

FIGURE 3 – Aperçu statistique du jeu de données

Ces statistiques fournissent un aperçu de la répartition et de la dispersion des différentes variables dans l'échantillon étudié :

Grossesses (Pregnancies) : La plupart des femmes dans l'échantillon ont en moyenne près de 4 grossesses, avec un minimum de 0 et un maximum de 17. Les quartiles suggèrent une distribution asymétrique, avec 50% des femmes ayant 3 grossesses ou moins.

Glucose : La concentration moyenne de glucose est d'environ 120.9 mg/dL, avec une dispersion relativement faible (écart-type de 32.04). Les valeurs de glucose varient de 0 à 199 mg/dL, avec la majorité des valeurs comprises entre 99 mg/dL (1er quartile) et 140.2 mg/dL (3ème quartile).

Pression artérielle (BloodPressure) : La pression artérielle diastolique moyenne est d'environ 69.11 mm Hg, avec une plage de 0 à 122 mm Hg. Les valeurs de pression artérielle semblent être réparties de manière relativement uniforme à travers l'échantillon.

Épaisseur du pli cutané (SkinThickness) : L'épaisseur moyenne du pli cutané au niveau du triceps est d'environ 20.54 mm. La dispersion des valeurs est assez large, allant de 0 à 99 mm, avec la moitié des valeurs comprises entre 0 et 32 mm.

Insuline (Insulin) : Le niveau moyen d'insuline sérique est d'environ 79.8 mu U/ml, avec une grande variabilité (écart-type de 115.24). Les valeurs d'insuline varient de 0 à 846 mu U/ml, avec 75% des valeurs inférieures à 127.2 mu U/ml.

Indice de masse corporelle (BMI) : L'indice de masse corporelle moyen est d'environ 31.99 kg/m^2, avec une gamme de 0 à 67.1 kg/m^2. La plupart des valeurs d'IMC se situent entre 27.30 kg/m^2 (1er quartile) et 36.60 kg/m^2 (3ème quartile).

Fonction du pedigree du diabète (DiabetesPedigreeFunction) : La valeur moyenne de la fonction du pedigree du diabète est d'environ 0.47, avec une dispersion considérable. Les valeurs varient de 0.078 à 2.42, avec une grande majorité des valeurs inférieures à 0.626.

Âge (Age) : L'âge moyen des individus est d'environ 33.24 ans, avec une plage de 21 à 81 ans. La distribution de l'âge semble relativement symétrique, avec la médiane proche de la moyenne.

Résultat (Outcome) : Environ 35% des individus ont développé le diabète, selon les données fournies.

1.5 Vérification des doublons

La présence de doublons peut affecter la validité des analyses et des résultats en introduisant des biais dans les données. En identifiant et en supprimant les doublons, nous nous assurons de maintenir l'intégrité et la fiabilité de notre jeu de données, ce qui est essentiel pour des analyses précises et robustes.

```r
# Check for duplicate values
check_duplicates <- function(data) {
  duplicates <- data[duplicated(data), ]
  if (nrow(duplicates) > 0) {
    print("Found duplicate rows:")
    print(duplicates)
    # Remove duplicates
    data <- data[!duplicated(data), ]
    print("Duplicates removed.")
  } else {
    print("No duplicate rows found.")
  }
  return(data)
}
```

```
[1] "No duplicate rows found."
```

FIGURE 4 – Vérification des doublons

Dans notre cas, après avoir effectué la vérification des doublons, nous constatons qu'aucun doublon n'a été trouvé. Cela confirme la cohérence et la fiabilité des données utilisées dans notre analyse, ce qui renforce la qualité de nos résultats et conclusions.

1.6 Vérification des valeurs manquantes

nous examinons la présence de valeurs manquantes dans notre jeu de données.

```r
# Function to check for missing values
check_missing_values <- function(data) {
  # Calculate the number and percentage of missing values for each column
  missing_values <- sapply(data, function(col) {
    sum(is.na(col))
  })
  missing_percentage <- sapply(data, function(col) {
    mean(is.na(col)) * 100
  })

  # Combine results into a data frame
  missing_summary <- data.frame(
    Column = names(missing_values),
    Missing_Values = missing_values,
    Missing_Percentage = round(missing_percentage, 2)
  )

  # Print the summary of missing values
  print("Missing Values Summary:")
  print(missing_summary)

  # Return the summary data frame
  return(missing_summary)
}

# Check for missing values in the dataset
missing_summary <- check_missing_values(db)
```

FIGURE 5 – Vérification des valeurs manquantes

```
[1] "Missing Values Summary:"
                                        Column Missing_Values Missing_Percentage
Pregnancies                          Pregnancies              0                  0
Glucose                                Glucose              0                  0
BloodPressure                    BloodPressure              0                  0
SkinThickness                    SkinThickness              0                  0
Insulin                                Insulin              0                  0
BMI                                        BMI              0                  0
DiabetesPedigreeFunction DiabetesPedigreeFunction              0                  0
Age                                        Age              0                  0
Outcome                                Outcome              0                  0
```

FIGURE 6 – affichage des valeurs manquantes

Dans notre jeu de données, nous constatons qu'aucune valeur n'est manquante (NULL). Cela indique que notre jeu de données est complet et prêt pour l'analyse, ce qui garantit la fiabilité de nos résultats.

1.7 Vérification des valeurs aberrantes.

Pour détecter les valeurs aberrantes, nous avons opté pour l'utilisation des boxplots, car ils offrent une représentation graphique de la distribution des données. Cette approche facilite l'identification des points de données qui se distinguent significativement de la majorité des données, permettant ainsi d'envisager leur statut potentiel d'outlier.

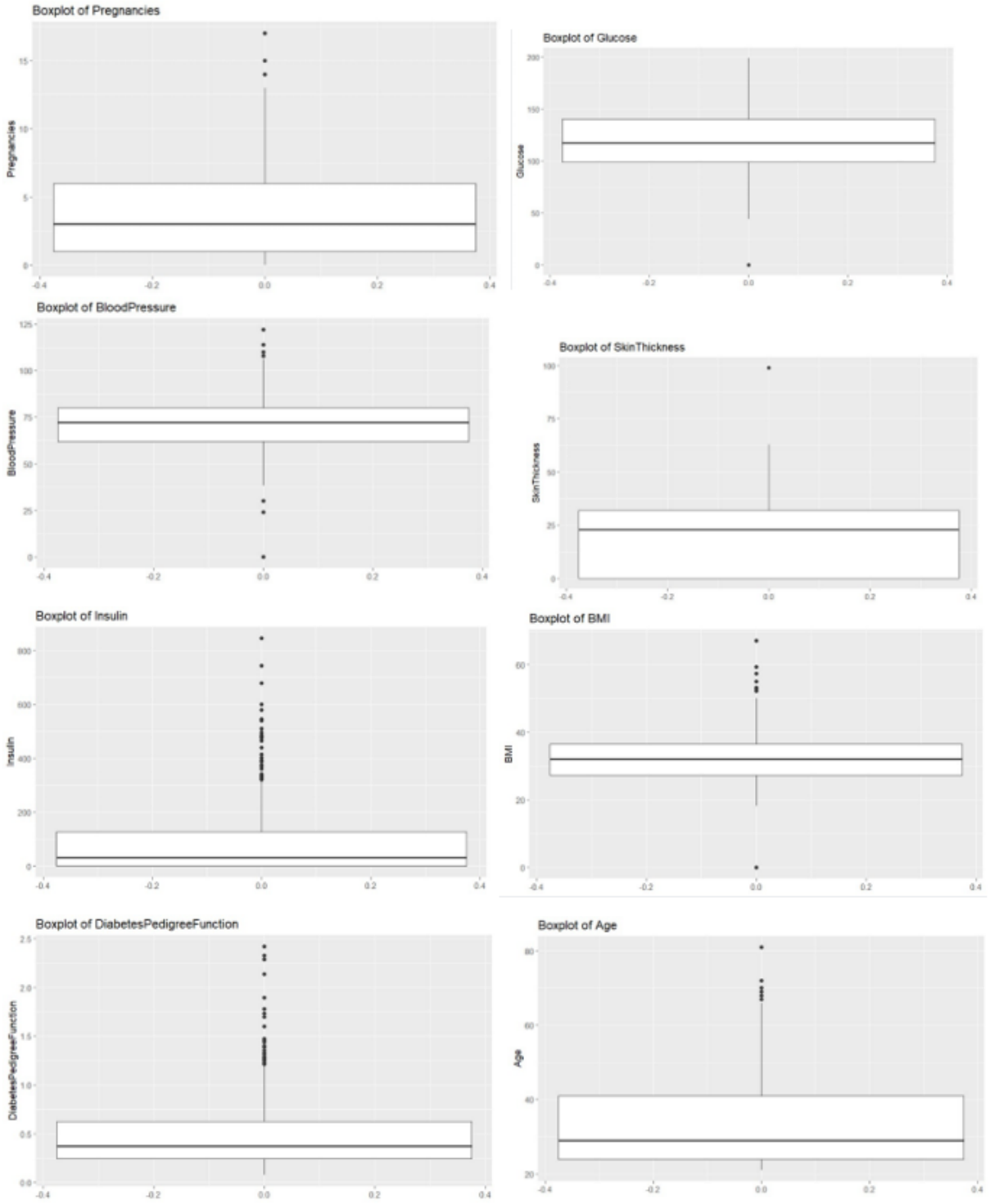


FIGURE 7 – Vérification des valeurs aberrantes

1.8 Exploration univariée.

Afin de mieux appréhender le jeu de données, nous avons représenté graphiquement la distribution des variables clés ainsi que les relations entre elles. De plus, une nouvelle colonne a été créée

pour enrichir l'analyse.

1.8.1 Création d'une Catégorie d'Âge

Dans le cadre de notre analyse, nous avons jugé pertinent de regrouper les individus en différentes catégories d'âge afin de mieux comprendre la répartition de notre population étudiée.

```
# Création de la variable catégorielle `Age_Cat` basée sur l'âge
db$Age_Cat <- cut(db$Age, breaks = c(0, 21, 26, 31, 36, 41, 51, 61, 120),
                  labels = c("<21", "21-25", "25-30", "30-35", "35-40", "40-50", "50-60", ">60"))
table(db$Age_Cat)
```

FIGURE 8 – création d'une catégorie d'âge

```
<21 21-25 25-30 30-35 35-40 40-50 50-60   >60
 63   237   141    73    82    99    48    25
```

FIGURE 9 – l'affichage du catégorie d'âge

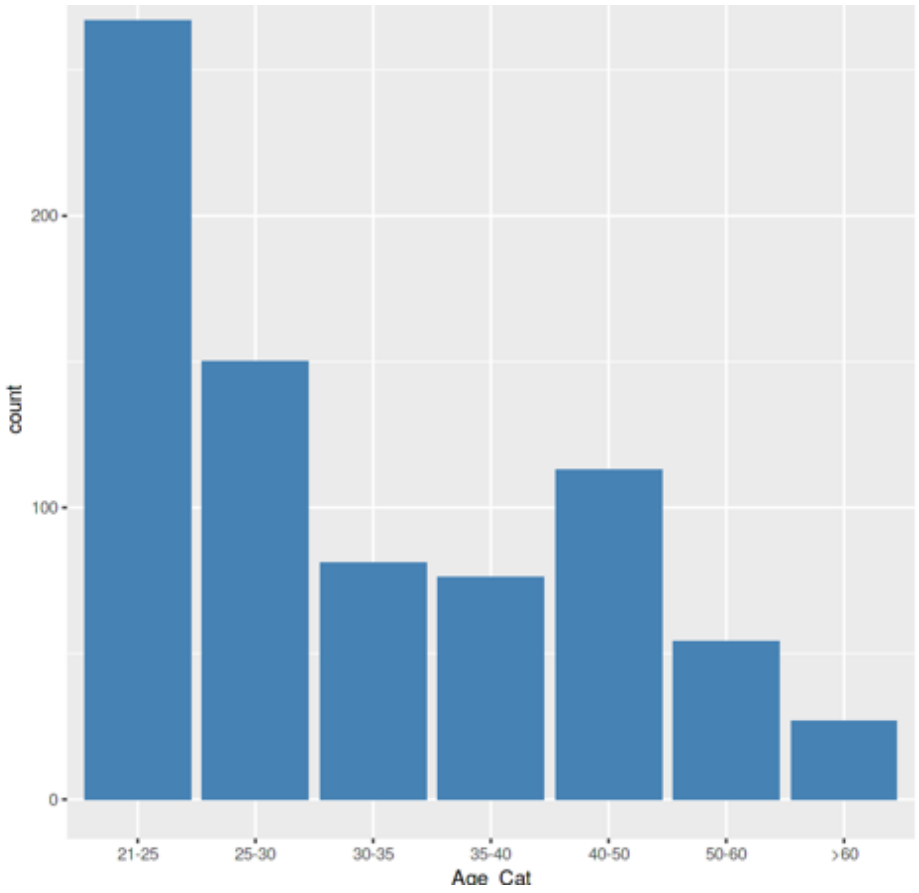


FIGURE 10 – La plupart des sujets ont un âge compris entre 21 et 30 ans.

1.8.2 Exploration des colonnes numériques :

SkinThickness :

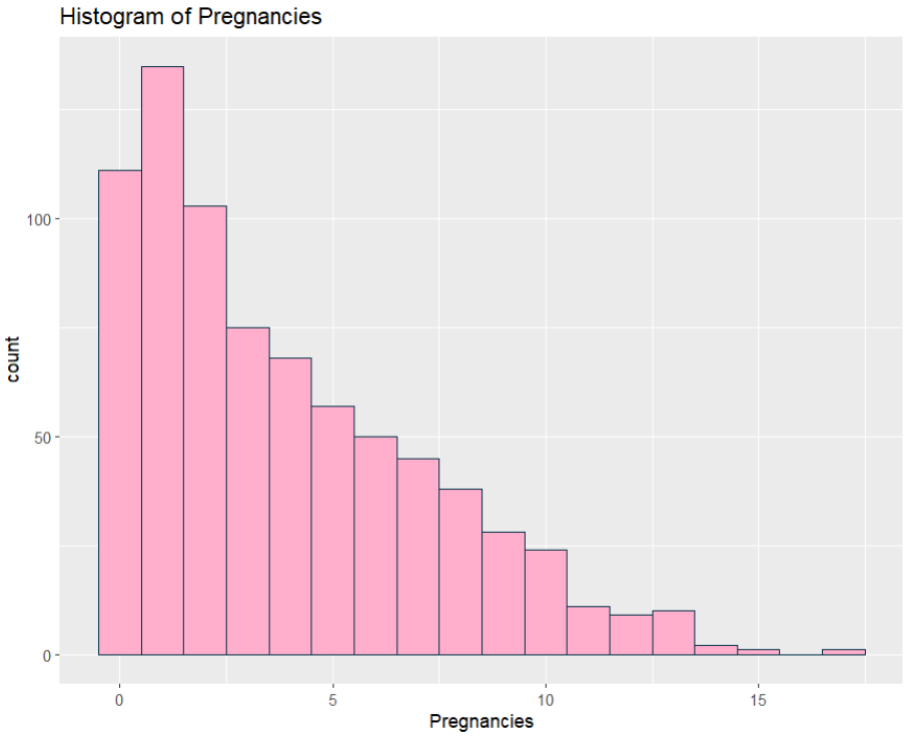


FIGURE 11 – SkinThickness

Glucose :

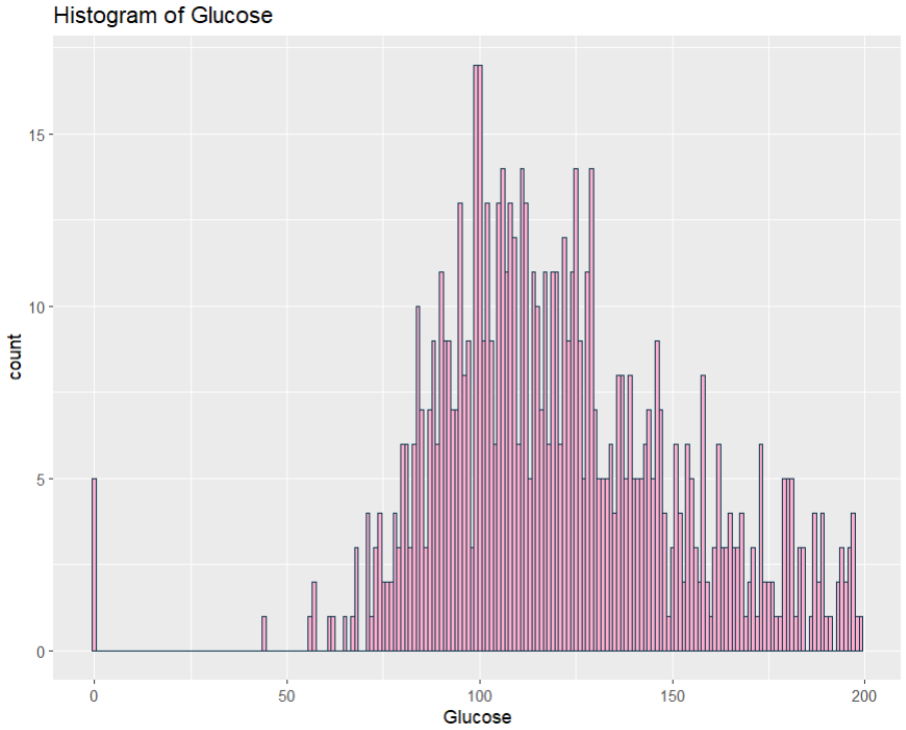


FIGURE 12 – Glucose

BloodPressure :

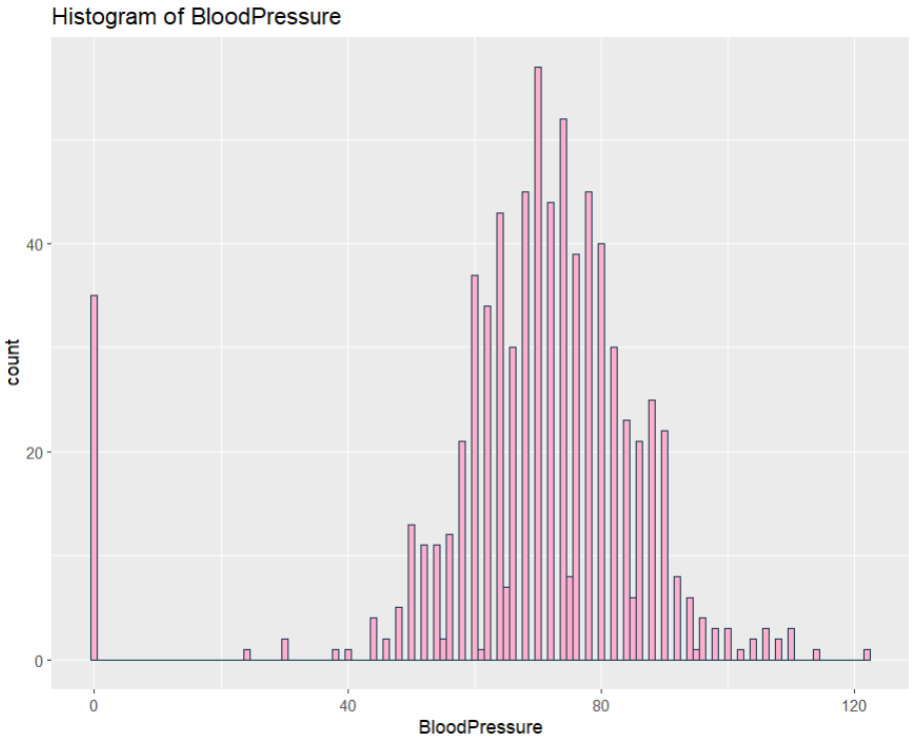


FIGURE 13 – BloodPressure

SkinThickness :

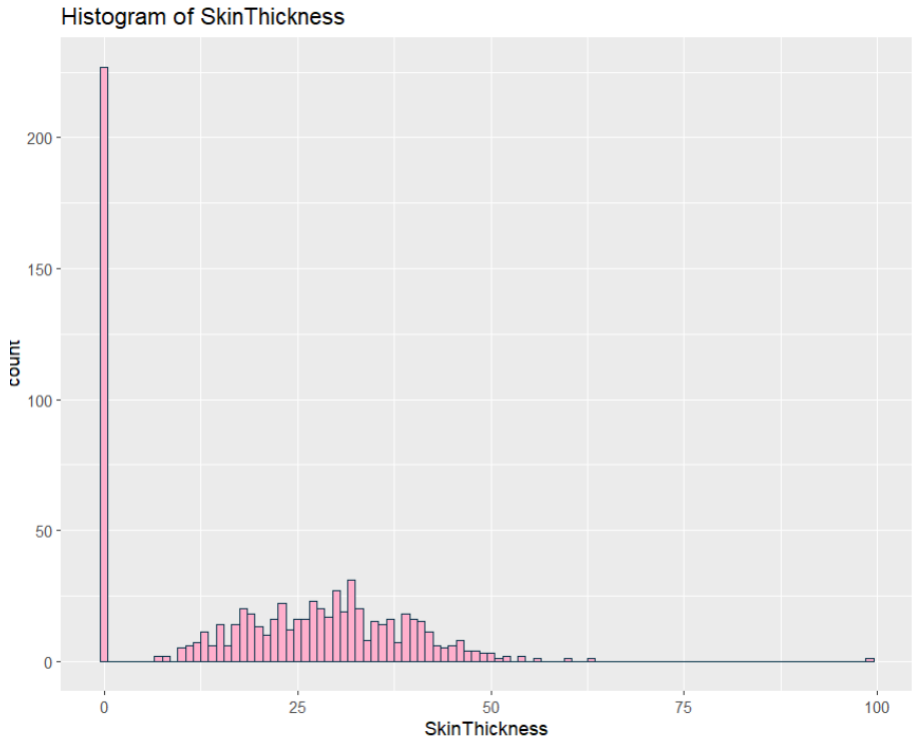


FIGURE 14 – SkinThickness

Insulin :

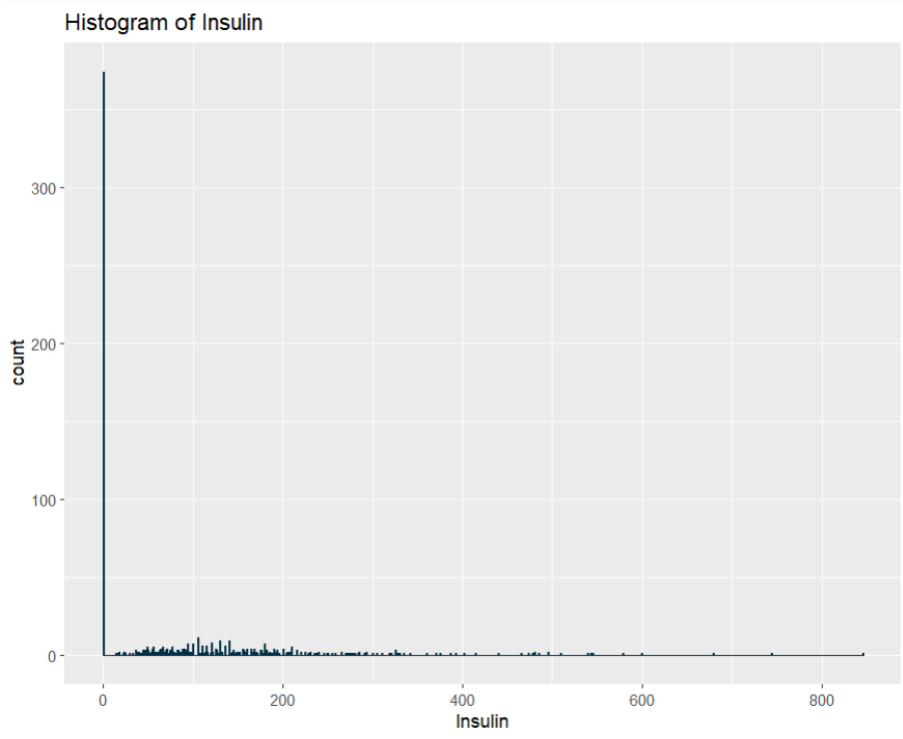


FIGURE 15 – Insulin

Indice de masse corporelle (BMI) :

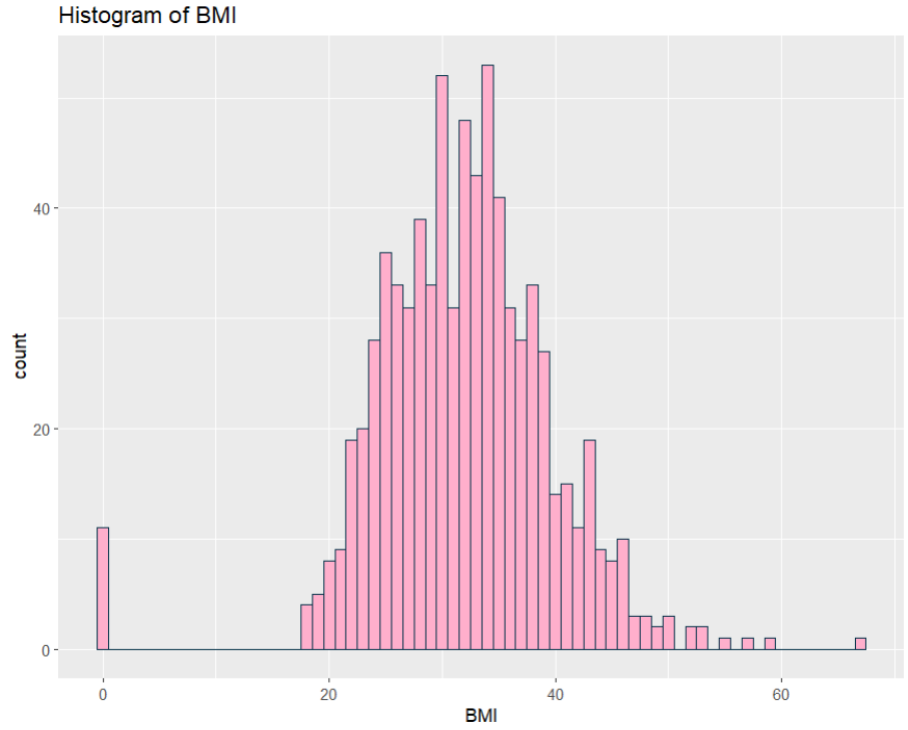


FIGURE 16 – BMI

DiabetesPedigreeFunction :

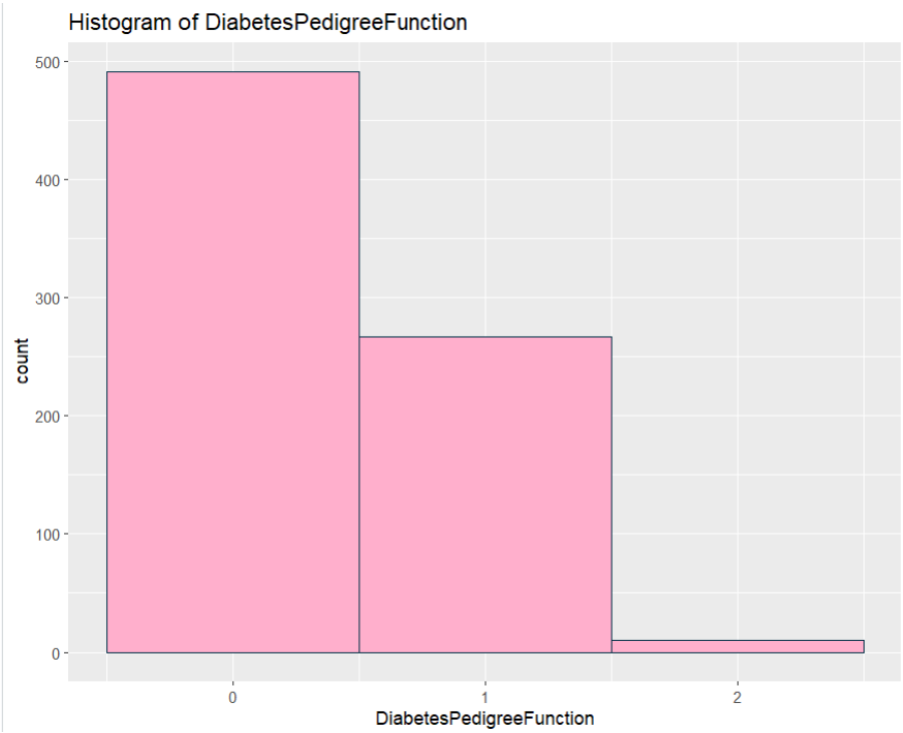


FIGURE 17 – DiabetesPedigreeFunction

Age :

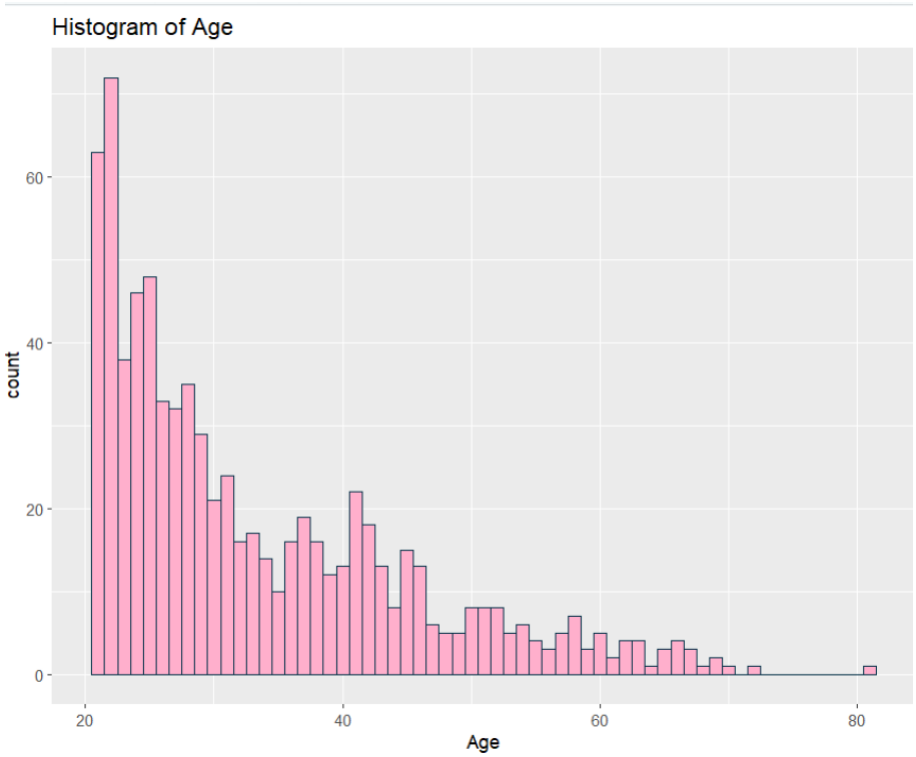
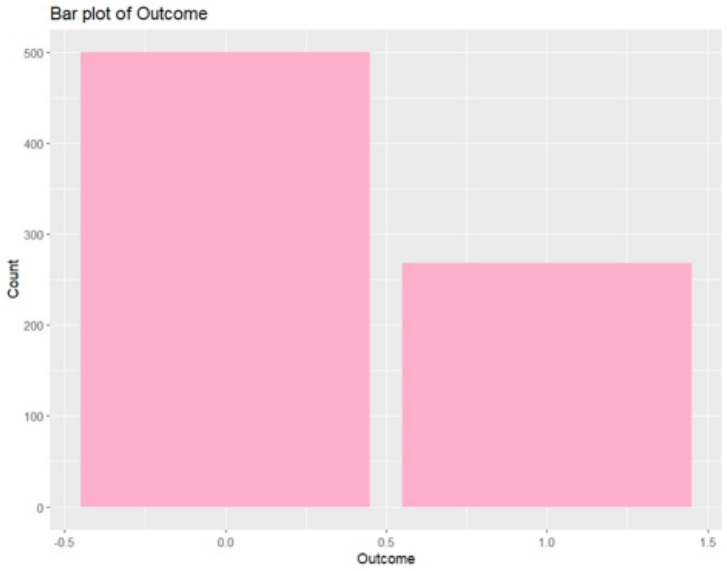


FIGURE 18 – Age

1.8.3 Exploration des colonnes catégorielles :

Outcome :




```
table(db$Outcome)

  0   1
500 268
```

FIGURE 19 – Outcome

Dans cette section, nous avons exploré visuellement toutes les colonnes catégorielles et numé-

riques de notre jeu de données. À travers les différents graphiques présentés, nous avons pu observer la répartition des valeurs pour chaque variable. Cette exploration nous a permis de comprendre la distribution des différentes catégories et de détecter d'éventuels schémas.

1.9 Analyse bivariée :

Dans cette étape d'analyse, nous étudierons la relation entre chaque paire de variables afin d'évaluer l'association entre une variable cible et une variable explicative. Notre objectif est de comprendre comment la variable cible varie en fonction de la variable explicative. Nous identifierons ainsi les variables les plus importantes et sélectionnerons celles qui sont pertinentes pour notre étude.

1.9.1 Exploration des relations : Catégorique Vs Continue – Boxplots

Dans cette section, nous utilisons des boxplots pour explorer visuellement les relations entre les variables catégorielles et continues. Cette méthode nous permet de comparer la distribution des valeurs continues pour chaque catégorie des variables catégorielles. Ces analyses nous aident à identifier les associations potentielles entre les variables et à comprendre la dynamique des données.

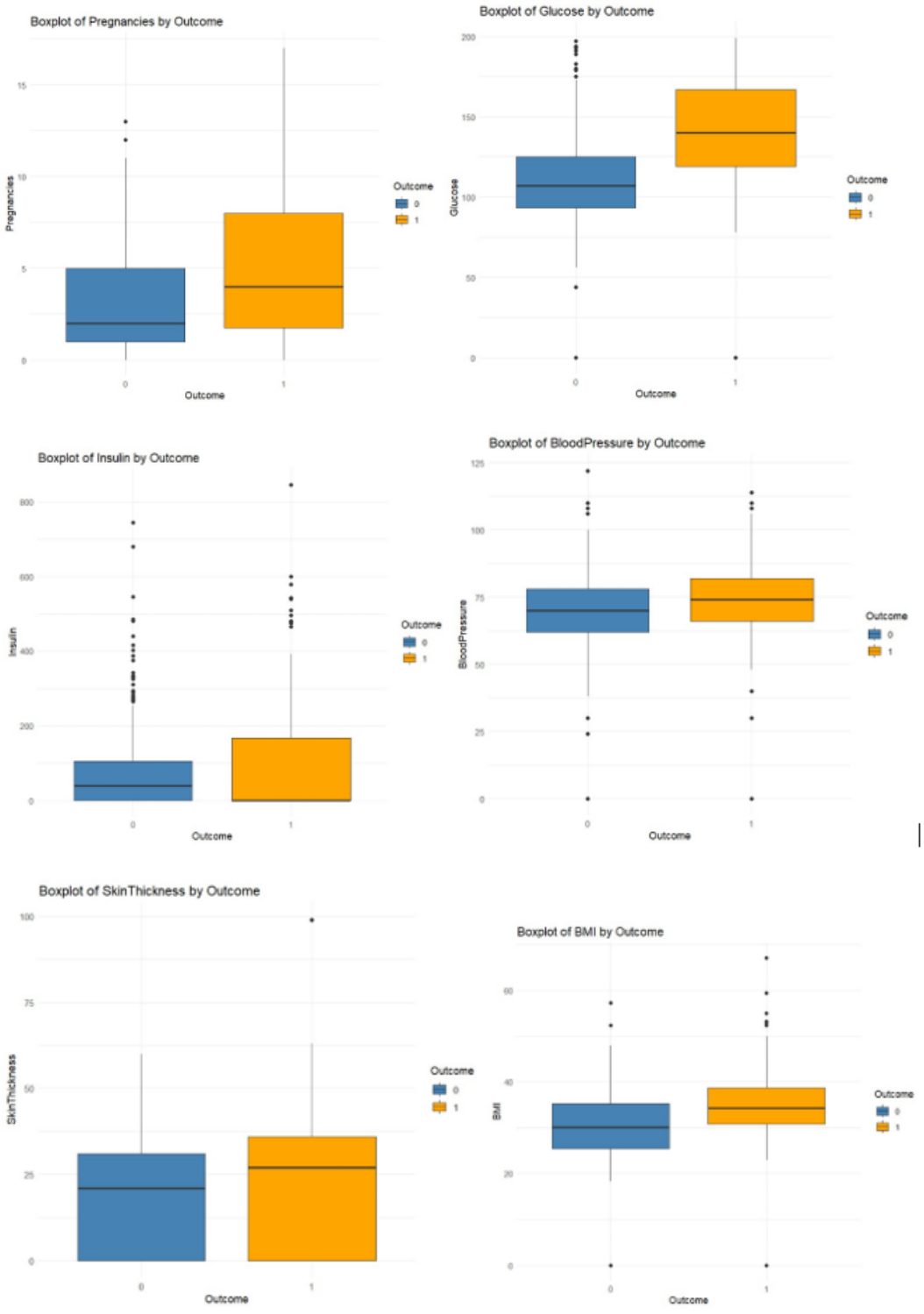


FIGURE 20 – Catégorique Vs Continue et Outcome (part 1)

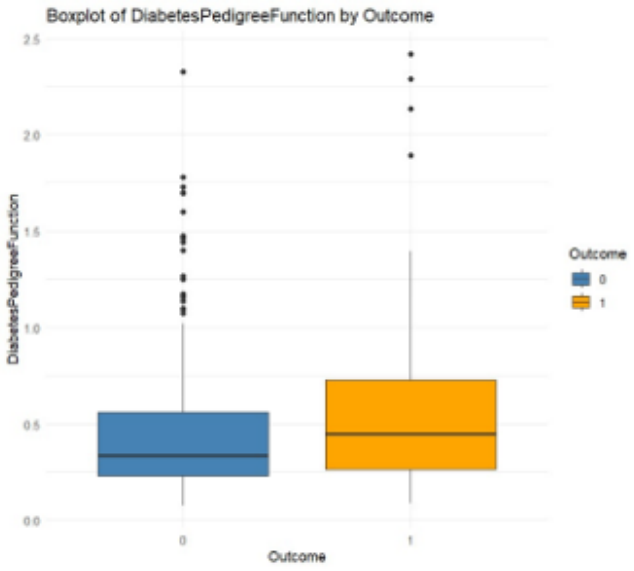


FIGURE 20 – Catégorique Vs Continue et Outcome (part 2)

Ces résultats présentent les statistiques descriptives pour différentes variables en fonction de la variable cible Outcome (0 pour pas de diabète, 1 pour diabète).

Pour la variable "Pregnancies", la moyenne et la médiane semblent plus élevées pour les individus ayant un diagnostic de diabète (Outcome = 1) par rapport à ceux sans diabète (Outcome = 0). Cela suggère une possible association entre le nombre de grossesses et le risque de diabète.

Pour la variable "Glucose", la moyenne et la médiane sont nettement plus élevées pour les individus avec diabète (Outcome = 1) par rapport à ceux sans diabète (Outcome = 0), ce qui est attendu car le glucose est un facteur de diagnostic clé pour le diabète.

Pour les autres variables telles que "BloodPressure", "SkinThickness", "Insulin", "BMI" et "DiabetesPedigreeFunction", on observe également des différences dans les statistiques descriptives entre les deux groupes d'Outcome, ce qui suggère des variations dans ces variables en fonction du statut du diabète.

Ces observations soulignent l'importance de ces variables dans la prédiction du diabète et justifient leur inclusion dans notre modèle d'analyse prédictive. Interprétation : Comme nous le savons, si la distribution semble similaire pour chaque catégorie (les boîtes sont alignées sur la même ligne), cela signifie que la variable continue n'a aucun effet sur la variable cible. Par conséquent, les variables ne sont pas corrélées entre elles. Mais ici, les distributions ne sont pas similaires.

1.9.2 Exploration des relations : Catégorique Vs Continue – Boxplots

Ces résultats montrent la distribution statistique des valeurs de l'indice de masse corporelle (BMI) pour chaque catégorie d'âge. Voici une interprétation pour chaque catégorie d'âge :
Pour les individus de moins de 21 ans (âge inférieur à 21), les valeurs de BMI ne sont pas disponibles.
Pour les individus âgés de 21 à 25 ans, la distribution du BMI a une médiane de 30.20 et une moyenne de 30.36
. Pour les individus âgés de 25 à 30 ans, la médiane du BMI est de 33.00 et la moyenne est de 33.04.
Pour les individus âgés de 30 à 35 ans, la médiane du BMI est de 32.00 et la moyenne est de 32.81.
Pour les individus âgés de 35 à 40 ans, la médiane du BMI est de 32.60 et la moyenne est de 32.97.
Pour les individus âgés de 40 à 50 ans, la médiane du BMI est de 33.8 et la moyenne est de 34.5.
Pour les individus âgés de 50 à 60 ans, la médiane du BMI est de 32.65 et la moyenne est de 31.11.
Pour les individus de plus de 60 ans (âge supérieur à 60), la médiane du BMI est de 28.0 et la moyenne est de 28.4.

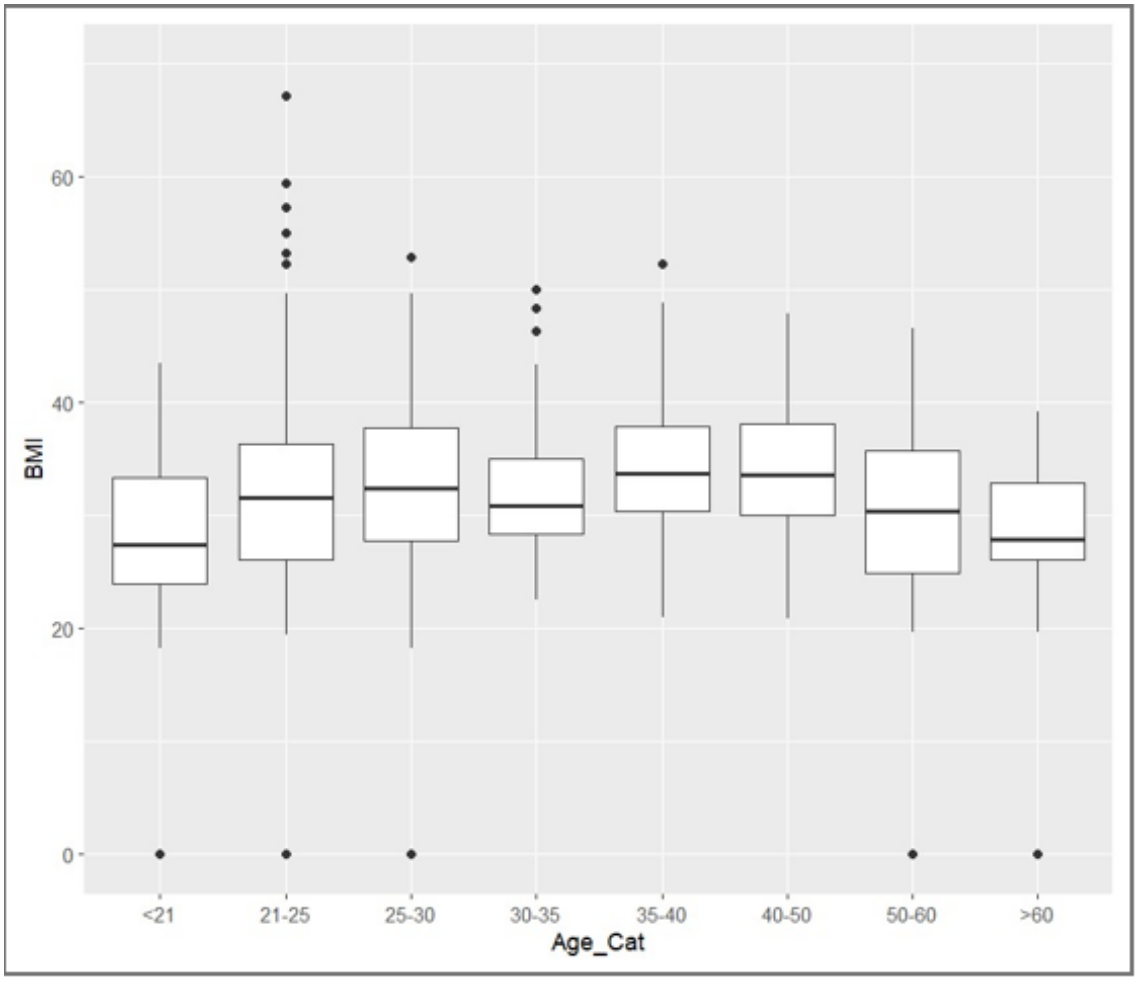


FIGURE 21 – Catégorique Vs Continue – Boxplots

1.9.3 Corrélation

Il est maintenant temps de sélectionner les meilleures colonnes (caractéristiques) qui sont corrélées à la variable cible. Cela peut être réalisé en mesurant directement les valeurs de corrélation. Nous avons donc effectué une analyse de corrélation pour les variables continues de notre jeu de données.

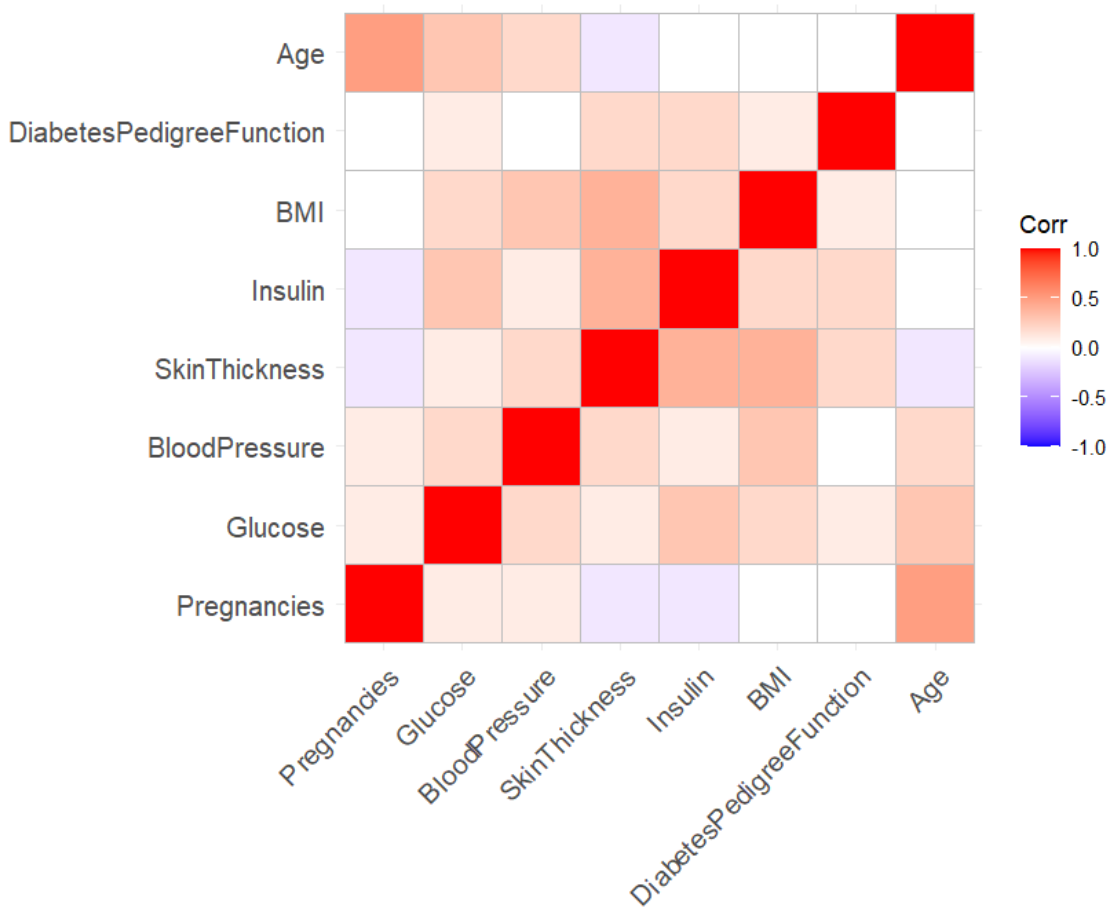


FIGURE 22 – Collinéarité des variables catégorielles

```
                          Pregnancies Glucose BloodPressure SkinThickness Insulin BMI
Pregnancies                       1.0     0.1           0.1          -0.1    -0.1 0.0
Glucose                           0.1     1.0           0.2           0.1     0.3 0.2
BloodPressure                     0.1     0.2           1.0           0.2     0.1 0.3
SkinThickness                    -0.1     0.1           0.2           1.0     0.4 0.4
Insulin                          -0.1     0.3           0.1           0.4     1.0 0.2
BMI                               0.0     0.2           0.3           0.4     0.2 1.0
DiabetesPedigreeFunction          0.0     0.1           0.0           0.2     0.2 0.1
Age                               0.5     0.3           0.2          -0.1     0.0 0.0
                          DiabetesPedigreeFunction  Age
Pregnancies                                    0.0  0.5
Glucose                                        0.1  0.3
BloodPressure                                  0.0  0.2
SkinThickness                                  0.2 -0.1
Insulin                                        0.2  0.0
BMI                                            0.1  0.0
DiabetesPedigreeFunction                       1.0  0.0
Age                                            0.0  1.0
> |
```

FIGURE 23 – Correlation Table

Interpretation : Une corrélation proche de 1 indique une forte corrélation positive entre les deux variables, tandis qu'une corrélation proche de -1 indique une forte corrélation négative. Une

corrélation proche de 0 indique une faible corrélation entre les variables. "Pregnancies" semble faiblement corrélé avec "Glucose", "BloodPressure", et "Age". "Glucose" présente une corrélation modérée avec "Insulin" et "BMI". "BloodPressure" a une corrélation modérée avec "BMI". "SkinThickness" est modérément corrélé avec "Insulin" et "BMI". "Insulin" a une corrélation modérée avec "SkinThickness" et "BMI". "BMI" est modérément corrélé avec "Glucose", "BloodPressure", "SkinThickness", et "Insulin".

1.9.4 Test du chi-deux

```r
# Fonction pour effectuer le test du chi-carré entre deux variables catégorielles
chi_square_test <- function(data, var1, var2) {
  # Crée un tableau croisé entre les deux variables
  contingency_table <- table(data[[var1]], data[[var2]])

  # Effectue le test du chi-carré
  chi_test <- chisq.test(contingency_table)

  # Affiche les résultats du test
  print(paste("Test du chi-carré entre", var1, "et", var2))
  print(chi_test)
}
```

FIGURE 24 – Fonction du chi-carré

```
[1] "Test du chi-carré entre Outcome et Age_Cat"

        Pearson's Chi-squared test

data:  contingency_table
X-squared = 88.134, df = 7, p-value = 2.988e-16
```

FIGURE 25 – Affichage les résultats du test du chi-carré

1.10 Conclusion

En conclusion, le processus de prétraitement des données revêt une importance capitale dans toute analyse de données et démarche d'apprentissage automatique. En préparant méticuleusement et en nettoyant nos données, nous nous assurons de leur qualité et de leur cohérence, ce qui améliore considérablement la fiabilité et l'efficacité de nos modèles. Les techniques examinées dans ce chapitre, telles que le nettoyage des données et la sélection des variables, nous permettent de tirer pleinement parti du potentiel de nos données et d'obtenir des résultats précis et fiables. Dotés d'une base solide en prétraitement des données, nous sommes mieux armés pour aborder des tâches complexes d'analyse de données et prendre des décisions éclairées basées sur des données de qualité supérieure.

2 Chapiter : Application de la régression logistique

2.1 Introduction

Une fois le prétraitement des données terminé, la prochaine étape consiste à appliquer la régression logistique à l'ensemble de données. La régression logistique est une méthode puissante utilisée pour modéliser la relation entre une variable binaire cible et un ensemble de variables explicatives. Nous explorerons comment cette technique peut être utilisée pour prédire le risque de diabète chez les patients en se basant sur les caractéristiques médicales fournies dans notre ensemble de données. Nous commencerons par une introduction à la régression logistique, en expliquant ses principes de base et son fonctionnement. Ensuite, nous appliquerons cette méthode à notre ensemble de données pour construire un modèle prédictif et évaluer ses performances. Enfin, nous interpréterons les résultats obtenus et discuterons de l'importance de la régression logistique dans le domaine de la santé publique et de la médecine préventive.

2.2 Application

2.2.1 Division de l'ensemble de données en données d'entraînement et de test

Nous divisons notre ensemble de données en ensembles d'entraînement et de test afin de pouvoir évaluer les performances de notre modèle de régression logistique. Nous utilisons la fonction 'sample.split' du package 'caTools' pour répartir aléatoirement les données en fonction de la variable cible "Outcome", en veillant à ce que la répartition soit équilibrée entre les deux classes. Nous avons choisi de répartir les données de manière à ce que 75 % soient utilisées pour l'entraînement (train) et 25 % pour le test (test). Ensuite, nous utilisons cette répartition pour extraire les données d'entraînement et de test à partir de notre ensemble de données initial. Ce processus nous permettra de construire un modèle sur les données d'entraînement et de le tester sur les données de test pour évaluer sa capacité à généraliser aux données invisibles.

```
# Division des données en ensembles d'entraînement et de test
set.seed(3)
sample <- sample.split(db$Outcome, SplitRatio = 0.75)
train <- subset(db, sample == TRUE)
test <- subset(db, sample == FALSE)
```

FIGURE 26 – Splitting the Data

2.2.2 Division de l'ensemble de données en données d'entraînement et de test

```
> #Nombre denregistrement dans la dataset :
> nrow(db)
[1] 768
> #nombre de lignes dans le dataframe train :
> nrow(train)
[1] 576
> #nombre de lignes dans le dataframe test :
> nrow(test)
[1] 192
```

FIGURE 27 – Exploration des données d'entraînement et de test

2.2.3 Distribution des tranches d'âge dans l'ensemble de données d'entraînement

```
# distribution of Age category in Train set
table(train$Age_Cat)

<21 21-25 25-30 30-35 35-40 40-50 50-60   >60
 46   174   104    55    58    80    37    22
```

FIGURE 28 – Distribution des tranches d'âge

2.2.4 Structure de l'ensemble de données d'entraînement

```
'data.frame':   576 obs. of  10 variables:
 $ Pregnancies             : int  6 1 1 0 5 3 10 2 8 10 ...
 $ Glucose                 : int  148 85 89 137 116 78 115 197 125 168 ...
 $ BloodPressure           : int  72 66 66 40 74 50 0 70 96 74 ...
 $ SkinThickness           : int  35 29 23 35 0 32 0 45 0 0 ...
 $ Insulin                 : int  0 0 94 168 0 88 0 543 0 0 ...
 $ BMI                     : num  33.6 26.6 28.1 43.1 25.6 31 35.3 30.5 0 38 ...
 $ DiabetesPedigreeFunction: num  0.627 0.351 0.167 2.288 0.201 ...
 $ Age                     : int  50 31 21 33 30 26 29 53 54 34 ...
 $ Outcome                 : Factor w/ 2 levels "0","1": 2 1 1 2 1 2 1 2 2 2 ...
 $ Age_Cat                 : Factor w/ 8 levels "<21","21-25",..: 6 3 1 4 3 2 3 7 7 4 ...
```

FIGURE 29 – l'ensemble de données d'entraînement

2.2.5 Modèle de référence

La commande table(db$Outcome) génère une table de fréquence des différentes valeurs de la variable cible Outcome dans l'ensemble de données complet db

```
> table(db$Outcome)

  0   1
500 268
```

FIGURE 30 – resultat de Modèle de référence

2.2.6 Précision de base

```
> # Baseline accuracy
> baseline <- round(500/nrow(db),2)
> baseline
[1] 0.65
```

FIGURE 31 – Précision de base

Dans cette section, nous calculons la précision de base de notre modèle en utilisant la répartition des résultats de l'ensemble de données. La précision de base est déterminée en divisant le nombre d'occurrences de la classe majoritaire par le nombre total d'observations dans l'ensemble de données. En l'occurrence, la classe majoritaire est celle où Outcome est égal à 0, avec 500 occurrences, sur un total de 768 observations dans l'ensemble de données. Ainsi, la précision de base est calculée en divisant 500 par 768, ce qui donne environ 0.65, ou 65%. Cela signifie que si nous prédisons simplement que chaque observation appartient à la classe majoritaire, notre modèle aurait une précision de 65%.

2.2.7 Construction du modèle de régression logistique

```
> AllVar <- glm(Outcome ~ ., data = train, family = binomial)
> summary(AllVar)

Call:
glm(formula = Outcome ~ ., family = binomial, data = train)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -6.942514   1.427209  -4.864 1.15e-06 ***
Pregnancies               0.045333   0.039853   1.137  0.25534
Glucose                   0.033999   0.004221   8.055 7.96e-16 ***
BloodPressure            -0.015682   0.006287  -2.494  0.01262 *
SkinThickness             0.003505   0.007705   0.455  0.64921
Insulin                  -0.001941   0.001028  -1.889  0.05894 .
BMI                       0.071094   0.017528   4.056 4.99e-05 ***
DiabetesPedigreeFunction  0.980498   0.348720   2.812  0.00493 **
Age                      -0.031847   0.053958  -0.590  0.55504
Age_Cat21-25              0.400663   0.609980   0.657  0.51128
Age_Cat25-30              1.194289   0.729331   1.638  0.10152
Age_Cat30-35              1.752090   0.931511   1.881  0.05998 .
Age_Cat35-40              1.904278   1.157606   1.645  0.09997 .
Age_Cat40-50              2.570509   1.464774   1.755  0.07928 .
Age_Cat50-60              2.611686   2.000857   1.305  0.19180
Age_Cat>60                1.421203   2.539854   0.560  0.57578
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 745.11  on 575  degrees of freedom
Residual deviance: 542.26  on 560  degrees of freedom
AIC: 574.26

Number of Fisher Scoring iterations: 5
```

FIGURE 32 – onstruction du modèle de régression logistique

Dans cette section, nous avons construit notre modèle de régression logistique en utilisant la commande ‘glm()‘ avec la spécification ‘Outcome .‘. Cela signifie que nous utilisons toutes les variables disponibles dans notre ensemble de données d'entraînement pour prédire la variable cible "Outcome". Après avoir ajusté le modèle, nous avons affiché un résumé de celui-ci en utilisant la fonction ‘summary()‘. Le résumé fournit des informations telles que les coefficients estimés pour chaque variable, leurs erreurs standard, les valeurs z correspondantes et les p-values associées. Les coefficients estimés représentent l'effet de chaque variable sur la probabilité de développer le diabète. Les p-values indiquent si chaque coefficient est statistiquement significatif pour prédire le résultat. En utilisant ce modèle, nous pouvons évaluer l'importance de chaque variable et estimer la probabilité de développer le diabète en fonction des caractéristiques médicales fournies.

2.2.8 Prédiction des résultats sur l'ensemble de données d'entraînement

```
> PredictTrain <- predict(AllVar, type = "response")
> summary(PredictTrain)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.002506 0.111359 0.270400 0.348958 0.543976 0.987233
>
```

FIGURE 33 – Prédiction des résultats

nous utilisons notre modèle de régression logistique pour prédire les résultats sur l'ensemble de données d'entraînement. Nous utilisons la fonction predict() en spécifiant le type "response" pour obtenir les probabilités prédites. Ensuite, nous fournissons un résumé des probabilités prédites. Les statistiques fournies comprennent la valeur minimale, le premier quartile, la médiane, la moyenne, le troisième quartile et la valeur maximale des probabilités prédites. Cela nous donne un aperçu de la distribution des probabilités prédites pour développer le diabète dans notre ensemble de données d'entraînement.

2.2.9 Évaluation des prédictions moyennes par classe

```
> tapply(PredictTrain, train$Outcome, mean)
        0         1
0.2352069 0.5611812
```

FIGURE 34 – Évaluation des prédictions moyennes par classe

Ce code calcule les prédictions moyennes pour chaque classe de la variable cible dans l'ensemble de données d'entraînement. Cela nous donne un aperçu de la tendance du modèle à prédire les résultats positifs et négatifs, en fournissant des estimations moyennes des probabilités prédites pour chaque classe.

2.2.10 Construction de la matrice de confusion

La matrice de confusion compare les résultats réels avec les prédictions faites par le modèle. Elle est divisée en quatre catégories :

— Vrais négatifs (VN) : observations correctement prédites comme négatives.

— Faux positifs (FP) : observations incorrectement prédites comme positives.

— Faux négatifs (FN) : observations incorrectement prédites comme négatives.

— Vrais positifs (VP) : observations correctement prédites comme positives.

La sensibilité mesure la capacité du modèle à détecter les vrais positifs, tandis que la spécificité mesure la capacité du modèle à éviter les faux positifs. La sensibilité est calculée en divisant le nombre de vrais positifs par la somme des vrais positifs et des faux négatifs :

$$\text{Sensibilité} = \frac{\text{VP}}{\text{VP} + \text{FN}}$$

La spécificité est calculée en divisant le nombre de vrais négatifs par la somme des vrais négatifs et des faux positifs :

$$\text{Spécificité} = \frac{\text{VN}}{\text{VN} + \text{FP}}$$

Le seuil est le point de décision utilisé pour transformer les probabilités prédites en prédictions binaires. **Un seuil élevé** conduit à une sensibilité plus faible mais à une spécificité plus élevée, tandis que **un seuil bas** conduit à une sensibilité plus élevée mais à une spécificité plus faible.

2.2.11 Évaluation du modèle avec un seuil de 0.5

Un seuil de 0,5 est souvent utilisé par défaut, mais il peut être ajusté en fonction des préférences pour différents types d'erreurs.

```
> threshold_0.5 <- table(train$Outcome, PredictTrain > 0.5)
> threshold_0.5

    FALSE TRUE
  0   328   47
  1    84  117
> # Accuracy
> accuracy_0.5 <- round(sum(diag(threshold_0.5))/sum(threshold_0.5),2)
> sprintf("Accuracy is %s",accuracy_0.5)
[1] "Accuracy is 0.77"
> # Mis-classification error rate
> MC_0.5 <- 1-accuracy_0.5
> sprintf("Mis-classification error is %s",MC_0.5)
[1] "Mis-classification error is 0.23"
> sensitivity0.5 <- round(118/(83+118),2)
> specificity0.5 <- round(333/(333+42),2)
> sprintf("Sensitivity at 0.5 threshold: %s", sensitivity0.5)
[1] "Sensitivity at 0.5 threshold: 0.59"
> sprintf("Specificity at 0.5 threshold: %s", specificity0.5)
[1] "Specificity at 0.5 threshold: 0.89"
>
```

FIGURE 35 – Évaluation du modèle avec un seuil de 0.5

Dans ce code, nous construisons la matrice de confusion en utilisant une valeur de seuil de 0.5. La matrice de confusion compare les valeurs réelles avec les valeurs prédites par le modèle. Ensuite, nous calculons l'exactitude (accuracy) et le taux d'erreur de classification (mis-classification error

rate) du modèle. De plus, nous calculons la sensibilité et la spécificité du modèle à un seuil de 0.5. Ces mesures évaluent la performance du modèle en termes de sa capacité à prédire correctement les classes positives et négatives.

2.2.12 Évaluation du modèle avec un seuil de 0.7

```
> threshold_0.7 <- table(train$Outcome, PredictTrain > 0.7)
> threshold_0.7

    FALSE TRUE
  0   360   15
  1   118   83
> # Accuracy
> accuracy_0.7 <- round(sum(diag(threshold_0.7))/sum(threshold_0.7),2)
> sprintf('Accuracy is %s', accuracy_0.7)
[1] "Accuracy is 0.77"
> # Mis-classification error rate
> MC_0.7 <- 1-accuracy_0.7
> sprintf("Mis-classification error is %s",MC_0.7)
[1] "Mis-classification error is 0.23"
> sensitivity0.7 <- round(78/(123+78),2)
> specificity0.7 <- round(359/(359+16),2)
> sprintf("Sensitivity at 0.7 threshold: %s", sensitivity0.7)
[1] "Sensitivity at 0.7 threshold: 0.39"
> sprintf("Specificity at 0.7 threshold: %s", specificity0.7)
[1] "Specificity at 0.7 threshold: 0.96"
>
```

FIGURE 36 – Évaluation du modèle avec un seuil de 0.7

La matrice de confusion ci-dessus illustre les performances du modèle de régression logistique avec un seuil de 0,7 pour la prédiction du diabète. Voici un résumé des résultats :

- Précision : 0,76
- Taux d'erreur de classification : 0,24
- Sensibilité (à un seuil de 0,7) : 0,39
- Spécificité (à un seuil de 0,7) : 0,96

La précision de 0,76 indique que le modèle prédit correctement environ 76 % des échantillons. Le taux d'erreur de classification est d'environ 24 %. La sensibilité du modèle à ce seuil est de 0,39, ce qui signifie qu'il identifie correctement 39 % des échantillons positifs, tandis que sa spécificité est de 0,96, indiquant qu'il identifie correctement 96 % des échantillons négatifs.

```
> threshold_0.2 <- table(train$Outcome, PredictTrain > 0.2)
> threshold_0.2

    FALSE TRUE
  0   210  165
  1    17  184
> # Accuracy
> accuracy_0.2 <- round(sum(diag(threshold_0.2))/sum(threshold_0.2),2)
> sprintf("Accuracy is %s", accuracy_0.2)
[1] "Accuracy is 0.68"
> # Mis-classification error rate
> MC_0.2 <- 1-accuracy_0.2
> sprintf("Mis-classification error is %s",MC_0.2)
[1] "Mis-classification error is 0.32"
> sensitivity0.2 <- round(180/(21+180),2)
> specificity0.2 <- round(215/(215+160),2)
> sprintf("Sensitivity at 0.2 threshold: %s",sensitivity0.2)
[1] "Sensitivity at 0.2 threshold: 0.9"
> sprintf("Specificity at 0.2 threshold: %s",specificity0.2)
[1] "Specificity at 0.2 threshold: 0.57"
>
```

FIGURE 37 – Évaluation du modèle avec un seuil de 0.2

2.2.13 Courbes ROC (Receiver Operator Characteristic Curve)

Les courbes ROC nous aident à déterminer le seuil optimal pour notre modèle.

- Seuil élevé :
  - Haute spécificité
  - Faible sensibilité
- Seuil faible :
  - Faible spécificité
  - Haute sensibilité

Les courbes ROC sont générées ci-dessus pour évaluer la performance du modèle. Les seuils sont marqués le long de la courbe, nous permettant de choisir le seuil optimal en fonction de nos besoins spécifiques. Plus l'AUC (aire sous la courbe) est proche de 1, meilleure est la performance du modèle.

```r
# Generate ROC Curves

library(ROCR)

ROCRpred = prediction(PredictTrain, train$Outcome)
ROCRperf = performance(ROCRpred, "tpr", "fpr")

# Adding threshold labels
plot(ROCRperf, colorize=TRUE, print.cutoffs.at = seq(0,1,0.1), text.adj = c(-0.2, 1.7))
abline(a=0, b=1)

auc_train <- round(as.numeric(performance(ROCRpred, "auc")@y.values),2)
legend(.8, .2, auc_train, title = "AUC", cex=1)


# Making predictions on test set

PredictTest <- predict(AllVar, type = "response", newdata = test)
```

FIGURE 38 – Courbes ROC

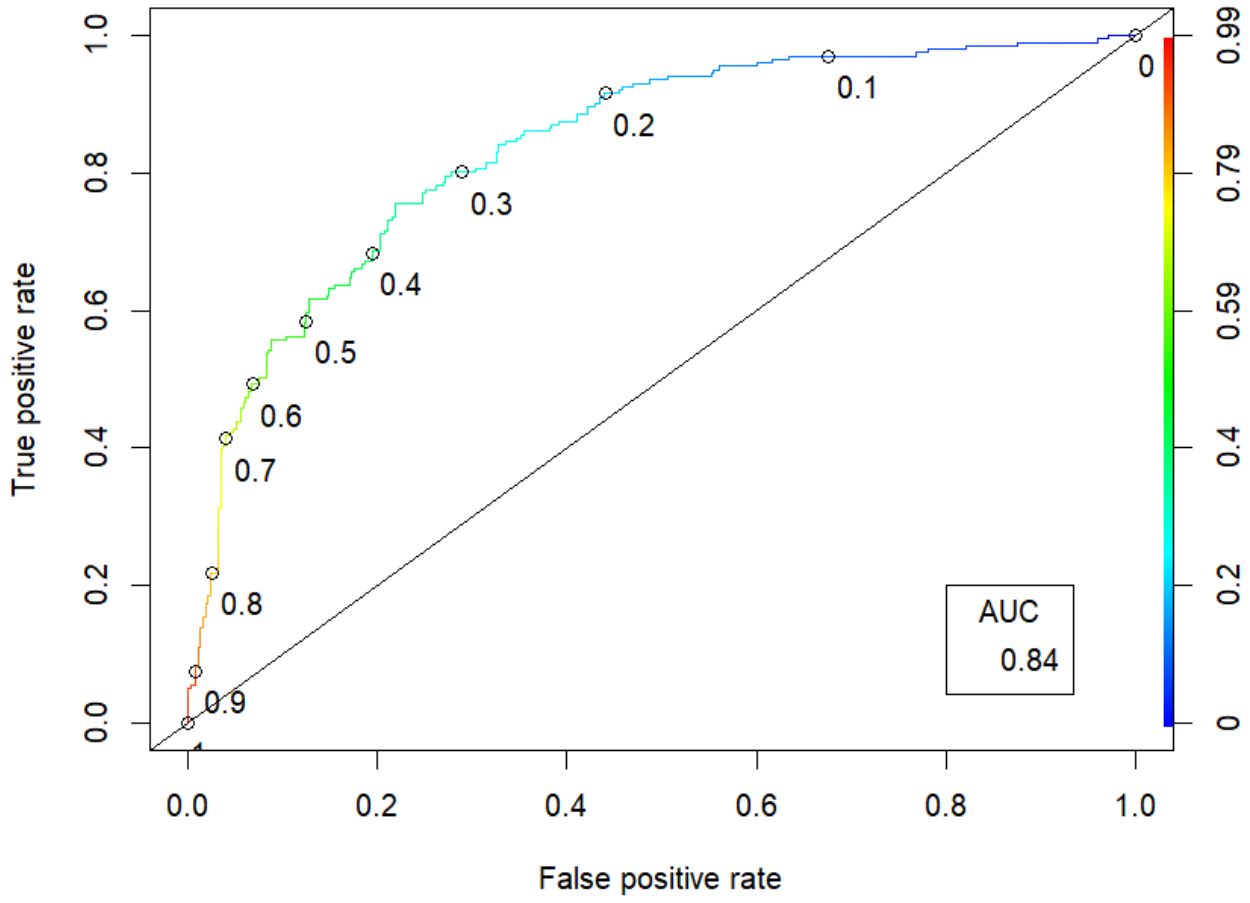


FIGURE 39 – resultat de Courbes ROC

2.2.14 Interprétation des résultats :

L'AUC sur l'ensemble de données d'entraînement est d'environ 0.85, ce qui suggère une bonne capacité de prédiction du modèle.

2.2.15 Interprétation du modèle :

L'AUC (Aire sous la courbe ROC) est une mesure qui indique la qualité de la prédiction d'un modèle. Les valeurs possibles de l'AUC sont les suivantes :

— AUC = 1 : Prédiction parfaite
— AUC = 0.5 : Prédiction aléatoire

Les différentes métriques utilisées pour évaluer la performance d'un modèle de classification binaire sont les suivantes :

— Classe prédite = 0, Classe réelle = 0 (Vrais Négatifs (TN))
— Classe prédite = 1, Classe réelle = 0 (Faux Positifs (FP))
— Classe prédite = 0, Classe réelle = 1 (Faux Négatifs (FN))
— Classe prédite = 1, Classe réelle = 1 (Vrais Positifs (TP))

Les métriques calculées à partir de ces valeurs sont les suivantes :

— Précision globale : $\frac{TN+TP}{N}$
— Sensibilité : $\frac{TP}{TP+FN}$
— Spécificité : $\frac{TN}{TN+FP}$
— Taux d'erreur global : $\frac{FP+FN}{N}$
— Taux d'erreur de faux négatifs : $\frac{FN}{TP+FN}$
— Taux d'erreur de faux positifs : $\frac{FP}{TN+FP}$

Le taux d'erreur de faux positifs est calculé comme suit : $1 -$ spécificité.

2.2.16 Prédictions sur l'ensemble de test

```
> ## Based on ROC curve above, selected a threshold of 0.5
> test_tab <- table(test$Outcome, PredictTest > 0.5)
> test_tab

    FALSE TRUE
  0   113   12
  1    27   40
> accuracy_test <- round(sum(diag(test_tab))/sum(test_tab),2)
> sprintf("Accuracy on test set is %s", accuracy_test)
[1] "Accuracy on test set is 0.8"
```

FIGURE 40 – Prédictions sur l'ensemble de test

L'exactitude sur l'ensemble de test est de 0.82, ce qui signifie que notre modèle prédit correctement environ 82 % des cas dans l'ensemble de test.

Calcul de l'AUC sur l'ensemble de test Nous avons calculé l'AUC (aire sous la courbe ROC) sur l'ensemble de test pour évaluer la capacité prédictive du modèle.

```
> ROCRPredTest = prediction(PredictTest, test$Outcome)
> auc = round(as.numeric(performance(ROCRPredTest, "auc")@y.values),2)
> auc
[1] 0.89
```

FIGURE 41 – Prédictions sur l'ensemble de test

L'AUC sur l'ensemble de test est de 89% , ce qui indique que la capacité prédictive du modèle est bonne.

2.3 Conclusion

Dans ce chapitre, nous avons exploré l'application de la régression logistique à notre ensemble de données. Après avoir effectué une analyse exploratoire approfondie et prétraité nos données, nous avons construit un modèle de régression logistique pour prédire le risque de diabète chez les patients. En utilisant des techniques telles que la division des données en ensembles d'entraînement et de test, la construction du modèle et l'évaluation de ses performances, nous avons pu obtenir des résultats prometteurs.

L'évaluation du modèle sur l'ensemble de test a révélé une précision de **82%** et une AUC de **0.89%**, ce qui suggère que le modèle est capable de prédire avec succès le risque de diabète chez les patients en se basant sur les caractéristiques médicales fournies dans notre ensemble de données.

En conclusion, ce chapitre a démontré l'efficacité de la régression logistique dans la modélisation de la relation entre les variables explicatives et la variable cible binaire. Ces résultats encouragent une exploration plus poussée de la régression logistique et d'autres techniques d'apprentissage automatique pour améliorer la prédiction et la compréhension du risque de diabète chez les patients.