ⵜⴰⴳⵍⴷⵉⵜ ⵓⵙⵏⵙ ⵎⵓⵃⵎⵎⴷ ⵉⴱⵏ ⵄⴱⴷⵍⵍⴰⵀ
ⵜⴰⵀⵙⵙⵍⵜ ⵜⴰⵎⴰⵡⴰⵙⵜ ⴻⴷⵀ ⵅⵎⵉⵙⴰⵅ

جامعة سيدي محمد بن عبدالله
كلية العلوم ظهر المهراز

Université Sidi Mohamed Ben Abdellah
Faculté des Sciences Dhar Mahraz
MASTER WISD

Prédiction de la pression sanguine avec la régression linéaire

Préparé par : AABBOU Zakaria
AMRANI Zouhir

Année universitaire: 2019/2020

SOMMAIRE

TABLE DE FIGURES

I. INTRODUCTION

La pression sanguine est la force exercée par le sang sur la paroi des vaisseaux, par unité de surface. En médecine, on parle de pression artérielle (ou de tension artérielle) pour désigner la pression mesurée dans les artères proches du cœur.

L'hypertension artérielle (HTA) est une maladie fréquente dans le monde. Liée à une pression anormalement élevée du sang dans les vaisseaux sanguins, elle semble anodine car elle généralement silencieuse. Si elle n'est pas traitée pendant une période prolongée, il peut entraîner des complications médicales importantes telles qu'une crise cardiaque, un accident vasculaire cérébral ou une maladie rénale. Pour remédier à ce problème, on souhaite développer un modèle qui prédit la tension artérielle, basé sur des mesures de santé et des informations limitées sur le mode de vie des patients. L'objectif est d'utiliser ce modèle pour développer un portail patient interactif qui fournit une estimation de la tension artérielle d'un patient en fonction de ses paramètres de santé et de son mode de vie.

Pour cela on a des données de 1 475 patients collecté au cours des 12 derniers mois. Les données utilisé dans cette étude sont des données du monde réel collectées par 'U.S. Centers for Disease Control and Prevention' dans le cadre de 'National Health and Nutrition Examination Survey (NHANES)'. De nombreuses données issues de cette enquête sont disponibles via le package RNHANES.

Les variables de notre base de données sont définies comme suit:

- **Systolic :** C'est la pression artérielle systolique (ou systole) du patient. L'unité de mesure est le millimètre de mercure (mmHg). C'est la variable dépendante que nous voulons prédire.
- **Weight :** C'est le poids mesuré du patient en kilogrammes (kg).
- **Height :** C'est la taille mesurée du patient en centimètres (cm).
- **Bmi :** C'est l'indice de masse corporelle du patient. Cela donne une idée de l'insuffisance pondérale ou le surpoids d'un patient.
- **Waist :** C'est la circonférence mesurée de la taille d'un patient en centimètres (cm).
- **Age :** C'est l'âge déclaré par le patient.
- **Diabetes :** C'est un indicateur binaire indiquant si le patient est diabétique (1) ou non (0).
- **Smoker :** C'est un indicateur binaire indiquant si le patient fume des cigarettes régulièrement (1) ou non (0).
- **Fastfood :** C'est un décompte déclaré du nombre de 'fastfood' que le patient a pris au cours d'une semaine écoulée.

II. ENSEMBLE DES PACKAGES ET DONNEES

1. Ensemble des packages

➢ **Tidyverse :**

Tidyverse est une collection de packages R conçus pour faciliter l'ensemble du processus d'analyse en offrant un format standardisé pour l'échange de données entre les packages. Il comprend des packages conçus pour importer, manipuler, visualiser et modéliser des données avec une série de fonctions qui fonctionnent facilement dans différents packages de tidyverse.

Voici les principaux packages qui composent le tidyverse:

- **readr** pour importer des données dans R à partir de divers formats de fichiers
- **tibble** pour stocker les données dans un format standardisé
- **dplyr** pour manipuler les données
- **ggplot2** pour la visualisation des données
- **tidyr** pour transformer les données en une forme «ordonnée»
- Et autres …

On peut facilement installer tous les packages *tidyverse* avec la commande:

```
1. install.packages("tidyverse")
```

➢ **Olsrr :**

Le package *olsrr* fournit les outils suivants pour l'enseignement et l'apprentissage de la régression linéaire à l'aide de R:

- Sortie de régression complète
- Diagnostics résiduels
- Mesures de l'influence
- Tests d'hétéroscédasticité
- Diagnostic de colinéarité
- Évaluation de l'ajustement du modèle
- Évaluation de la contribution variable
- Procédures de sélection de variables

Le package *olsrr* peut être installer avec la commande suivante:

```
1. install.packages("olsrr")
```

2. Importation des données

Nous commençons par l'importation de nos données en utilisant la fonction *read_csv()* du paquet *tidyverse*.

```
1. library (tidyverse)
2. health <- read_csv ("health.csv")
```

Nous avons importé avec succès les 1 475 observations et 9 variables. Pour obtenir un aperçu rapide de nos données, nous utilisons la commande *glimpse()* pour nous montrer les noms des variables, les types des données et quelques exemples de données.

```
1. glimpse(health)
```

```
Observations: 1,475
Variables: 9
$ systolic <dbl> 100, 112, 134, 108, 128, 102, 126, 124, 166, 138, 118, 124, 96, 116,...
$ weight   <dbl> 98.6, 96.9, 108.2, 84.8, 97.0, 102.4, 99.4, 53.6, 78.6, 135.5, 72.3,...
$ height   <dbl> 172.0, 186.0, 154.4, 168.9, 175.3, 150.5, 157.8, 162.4, 156.9, 180.2...
$ bmi      <dbl> 33.3, 28.0, 45.4, 29.7, 31.6, 45.2, 39.9, 20.3, 31.9, 41.7, 28.6, 31...
$ waist    <dbl> 120.4, 107.8, 120.3, 109.0, 111.1, 130.7, 113.2, 74.6, 102.8, 138.4,...
$ age      <dbl> 43, 57, 38, 75, 42, 63, 58, 26, 51, 61, 47, 52, 64, 55, 72, 80, 71, ...
$ diabetes <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,...
$ smoker   <dbl> 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0,...
$ fastfood <dbl> 5, 0, 2, 1, 1, 3, 6, 5, 0, 1, 0, 3, 0, 1, 0, 5, 0, 2, 1, 3, 2, 0, 12...
```

Comme nous l'avons vu précédemment, la variable ‘systolic’ sera la variable qu’on souhaite a expliqué et les autres variables seront des variables explicatives. Il est à noter que toutes les variables ont été importées sous forme numérique. Cependant, nous savons que les variables ‘diabetes’ et ‘smoker’ sont en fait des valeurs catégorielles. Nous devons donc convertir ces variables en facteurs en utilisant la fonction *as.factor()*.

```
1. health$diabetes= as.factor(health$diabetes)
2. health$smoker=as.factor(health$smoker)
```

3. Explorer les données

Maintenant que nous avons nos données, on va les explorées. Nous commençons par utiliser la fonction *summary()* pour obtenir un résumé statistique des variables numériques de nos données.

```
1. summary(health)
```

```
   systolic         weight          height           bmi             waist      
 Min.   : 80.0   Min.   : 29.10   Min.   :141.2   Min.   :13.40   Min.   : 56.2  
 1st Qu.:114.0   1st Qu.: 69.15   1st Qu.:163.8   1st Qu.:24.10   1st Qu.: 88.4  
 Median :122.0   Median : 81.00   Median :170.3   Median :27.90   Median : 98.9  
 Mean   :124.7   Mean   : 83.56   Mean   :170.2   Mean   :28.79   Mean   :100.0  
 3rd Qu.:134.0   3rd Qu.: 94.50   3rd Qu.:176.8   3rd Qu.:32.10   3rd Qu.:109.5  
 Max.   :224.0   Max.   :203.50   Max.   :200.4   Max.   :62.00   Max.   :176.0  

      age        diabetes  smoker     fastfood      
 Min.   :20.00   0:1265    0:770   Min.   : 0.00  
 1st Qu.:34.00   1: 210    1:705   1st Qu.: 0.00  
 Median :49.00                     Median : 1.00  
 Mean   :48.89                     Mean   : 2.14  
 3rd Qu.:62.00                     3rd Qu.: 3.00  
 Max.   :80.00                     Max.   :22.00
```

En regardant la distribution statistique de la variable 'systolic', nous voyons que la moyenne et la médiane sont relativement proches, ce qui suggère que les données sont normalement distribuées. En utilisant un histogramme, nous pouvons obtenir une représentation visuelle de la distribution (Figure 1).

```
# Visualiser la variable systolic avec ggplot().
ggplot() +
  geom_histogram(mapping=aes(x=health$systolic), fill="lightblue", color="black") +
  theme_minimal() +
  theme(text = element_text(size=14))
```

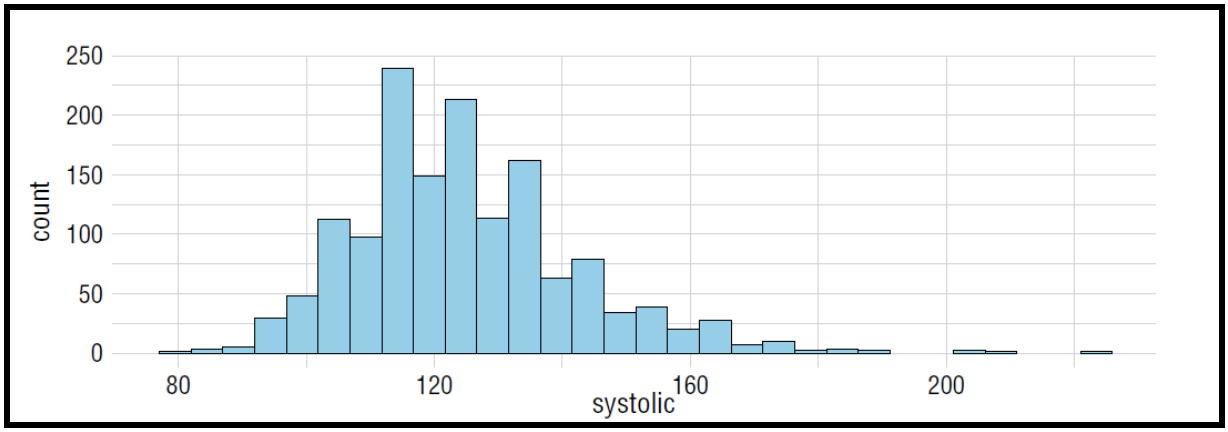

FIGURE 1 : DISTRIBUTION DES DONNEES DE LA VARIABLE 'SYSTOLIC'

L'histogramme montre que les données de la variable 'systolic' sont normalement distribuées.

De la même manier, on va examiner également les distributions statistiques des variables de prédiction à l'aide d'un ensemble d'histogrammes. Nous faisons cela en utilisant les fonctions 'tidyverse' *keep()*, *gather()* et *facet_wrap()* (Figure 2).

```
health %>%
select (-systolic) %>%
keep (is.numeric) %>%
gather () %>%
ggplot () +
geom_histogram(mapping = aes(x=value,fill=key), color = "black") +
facet_wrap (~ key, scales = "free") +
theme_minimal ()
```

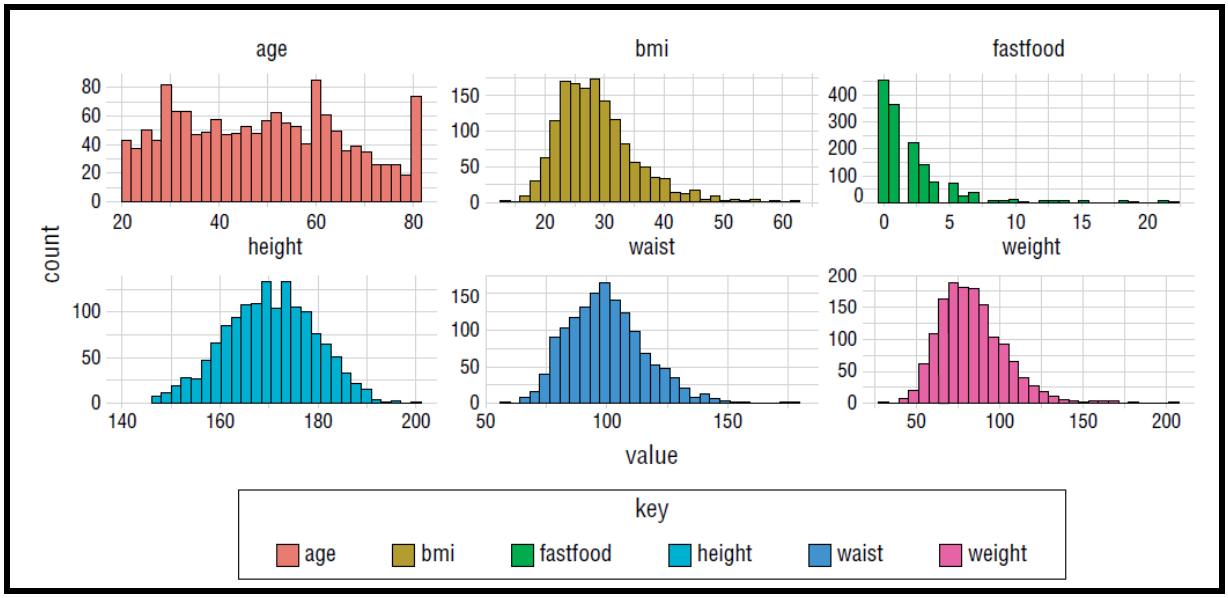

FIGURE 2 : DISTRIBUTIONS DES VARIABLES DANS LA BASE DE DONNEE

Nous voyons une distribution presque uniforme pour la variable 'age'. Cela signifie que nos données sont représentatives des patients à travers un large spectre d'âge, C'est à prévoir. La variable 'fastfood' est biaisée à droite. La plupart des patients consomment des fastfoods comme repas moins de cinq fois par semaine. Le reste de nos variables sont normalement distribué. Avec une aperçue visuelle, on constate qu'il n'y a pas de valeurs aberrantes évidentes (outliers) dans nos données qu'on va traiter.

L'étape suivante que nous devons faire dans le cadre du processus d'exploration des données est d'examiner la corrélation entre nos variables continues. Pour ce faire, nous utilisons la fonction *pairs()* qui va nous permet d'avoir une représentation graphique da la variable 'systolic' en fonction des autres variables.

```
1. #représentation graphique des corrélations
2. pairs(systolic~weight+height+bmi+waist+age,data=health , pch=20)
```

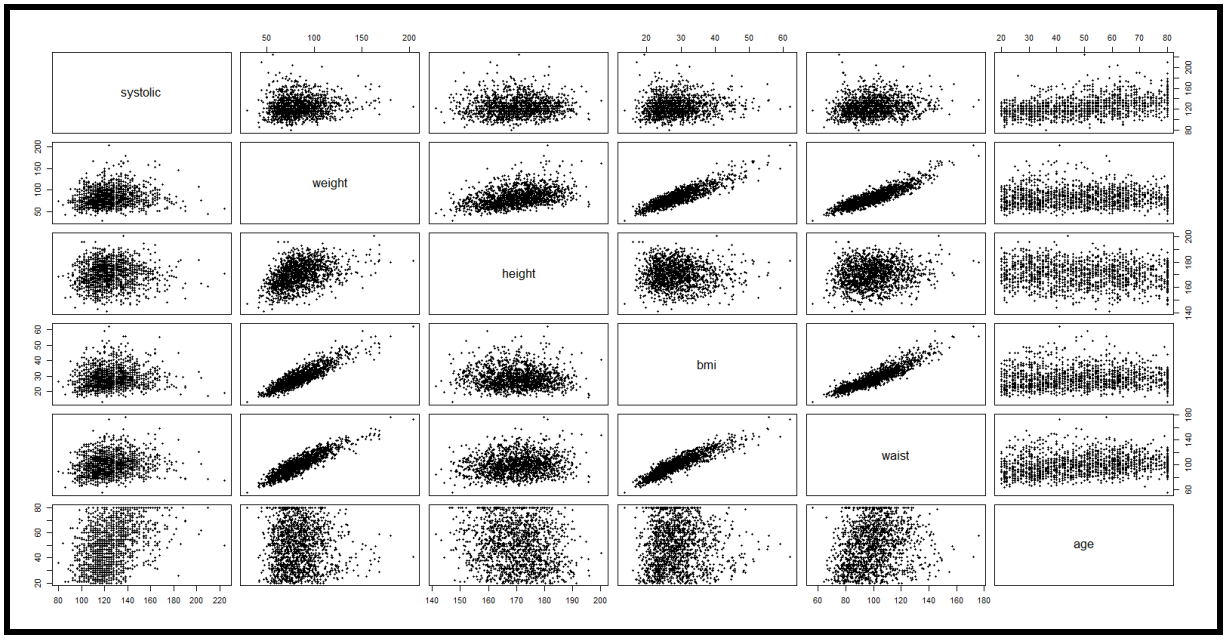

FIGURE 3 : REPRESENTATION GRAPHIQUE DES CORRELATIONS

Et pour avoir les valeurs des corrélations entre les variables, on va utiliser la fonction *cor()*

```
# Corrélation entre les variables continues.
cor(health[c("systolic","weight","height","bmi","waist","age","fastfood")])
```

```
            systolic      weight      height         bmi       waist         age    fastfood
systolic  1.00000000  0.10021386  0.02301030  0.09054668  0.16813021  0.40170911 -0.08417538
weight    0.10021386  1.00000000  0.40622019  0.89152826  0.89928820 -0.02217221  0.05770725
height    0.02301030  0.40622019  1.00000000 -0.03848241  0.14544676 -0.12656952  0.10917107
bmi       0.09054668  0.89152826 -0.03848241  1.00000000  0.91253710  0.03379844  0.01003525
waist     0.16813021  0.89928820  0.14544676  0.91253710  1.00000000  0.19508769 -0.02167324
age       0.40170911 -0.02217221 -0.12656952  0.03379844  0.19508769  1.00000000 -0.30089756
fastfood -0.08417538  0.05770725  0.10917107  0.01003525 -0.02167324 -0.30089756  1.00000000
```

En regardant la colonne 'systolic', nous pouvons voir que le prédicteur d'âge a la corrélation la plus forte avec la pression artérielle systolique. Viennent ensuite le tour de taille et le poids, tous deux faiblement corrélés. Il est intéressant de noter la corrélation négative entre la consommation de fastfood et la pression artérielle systolique. Cela semble inhabituel et contre-intuitif; cependant, la corrélation négative est assez faible et n'aura donc pas d'impact significatif sur notre modèle.

III. LES MODELES DE REGRESSION LINIERE

1. Ajustement du modèle de régression linéaire simple

De notre exploration, nous avons découvert que le prédicteur d'âge a la plus forte corrélation avec la variable 'systolic'. Nous allons donc commencer par construire un modèle de régression linéaire simple en utilisant l'âge comme prédicteur et la variable 'systolic' comme réponse.

```
health_mod1 <- lm (data=health, systolic~age)
summary (health_mod1)
```

```
Call:
lm(formula = systolic ~ age, data = health)

Residuals:
    Min      1Q  Median      3Q     Max
-42.028 -10.109  -1.101   8.223  98.806

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 104.34474    1.28169   81.41   <2e-16 ***
age           0.41698    0.02477   16.84   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.14 on 1473 degrees of freedom
Multiple R-squared:  0.1614,  Adjusted R-squared:  0.1608
F-statistic: 283.4 on 1 and 1473 DF,  p-value: < 2.2e-16
```

Les résultats montrent que les prédicteurs sont significatifs. Le coefficient d'âge nous dit que pour chaque augmentation de 0,4 an de l'âge d'un patient, il faut s'attendre à ce que sa tension artérielle systolique augmente de 1 point. Cela signifie qu'en moyenne, plus un patient est âgé, plus sa tension artérielle est élevée. En d'autres termes on peut écrire :

$$systolic = 104.34 + 0.41 \times (age)$$

On présente le résultat sous forme d'un graphique on peut utiliser le code suivant :

```
1. ggplot(health, aes(x = age, y = systolic)) +
2.   geom_point() +
3.   stat_smooth(method = "lm", col = "red")
```

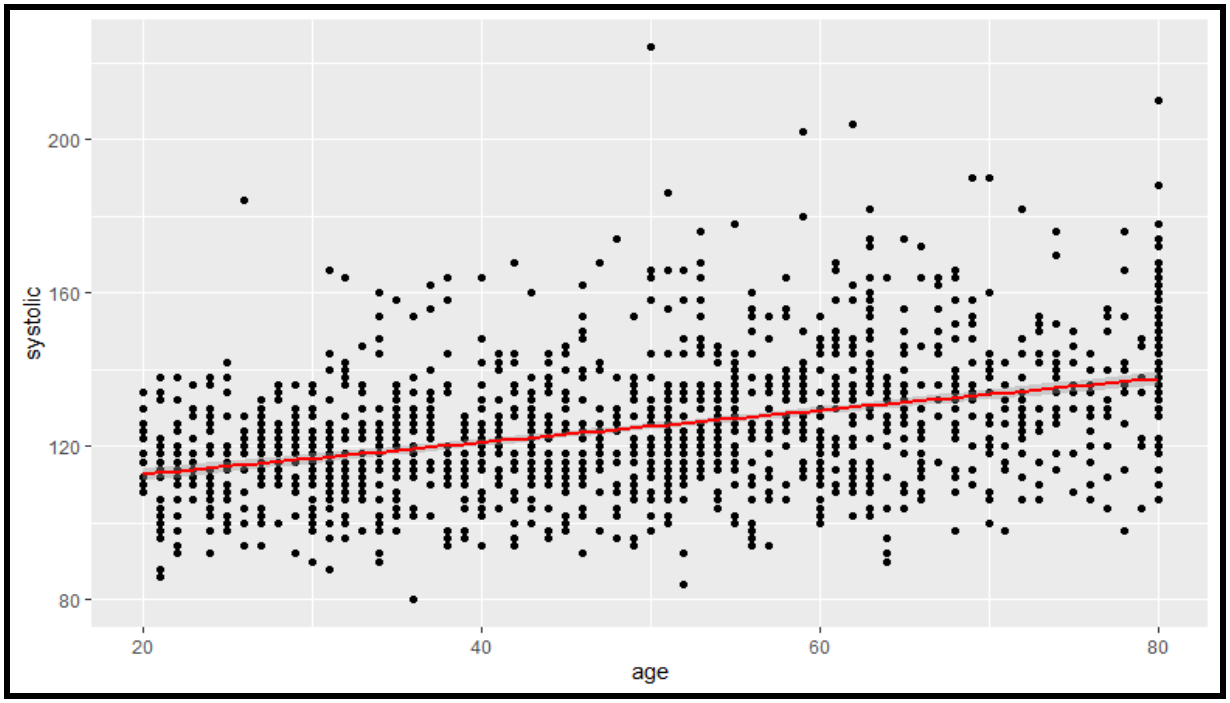

FIGURE 4 : LIGNE DE REGRESSION ESTIMEE ET VALEURS REELLES DE L'AGE EN FONCTION DE LA VARIABLE 'SYSTOILC'

On remarque bien la relation linéaire: la plupart des points semblent se concentrer autour de la ligne et quelques points sont éloigné de la ligne. Cependant, les points ne tombent jamais exactement sur la ligne droite. Sinon, le graphique représenterait une relation parfaite.

En regardant les diagnostics de modèle, nous voyons que l'erreur standard résiduelle est faible et la statistique F est statistiquement significative. Ce sont deux bons indicateurs de l'ajustement du modèle. Cependant, le R-carré nous indique que notre modèle n'explique que 16% de la variabilité de la réponse. Voyons si nous pouvons faire mieux en introduisant des prédicteurs supplémentaires dans le modèle.

2. Ajustement du modèle de régression linéaire multiple

Pour le cas de modèle de régression linéaire multiple, nous commencerons par tous les prédicteurs de nos données et par la variable 'systolic' comme réponse.

```
1. health_mod2 <- lm (data=health, systolic~.)
2. summary (health_mod2)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-41.463 -10.105  -0.765   8.148 100.398

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 163.30026   33.52545   4.871 1.23e-06 ***
weight        0.55135    0.19835   2.780  0.00551 **
height       -0.39201    0.19553  -2.005  0.04516 *
bmi          -1.36839    0.57574  -2.377  0.01759 *
waist        -0.00955    0.08358  -0.114  0.90905
age           0.43345    0.03199  13.549  < 2e-16 ***
diabetes1     2.20636    1.26536   1.744  0.08143 .
smoker1       1.13983    0.90964   1.253  0.21039
fastfood      0.17638    0.15322   1.151  0.24985
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.99 on 1466 degrees of freedom
Multiple R-squared:  0.1808,       Adjusted R-squared:  0.1763
F-statistic: 40.44 on 8 and 1466 DF,  p-value: < 2.2e-16
```

Les résultats montrent que les estimations de coefficient pour le poids, la taille, bmi, l'âge et le diabète sont significatives dans le modèle. Les diagnostics de modèle montrent également une légère réduction de l'erreur standard résiduelle, une légère augmentation du R-carré ajusté et une statistique F significative supérieure à 0. Généralement, ce modèle offre un meilleur ajustement que le modèle précédent. Exécutons maintenant des tests de diagnostic supplémentaires sur notre nouveau modèle.

Le premier test que nous effectuons est le test de la moyenne nulle des résidus.

```
1. mean (health_mod2$residuals)
```

```
[1] -1.121831e-15
```

La moyenne résiduelle est très proche de zéro, Alors, notre modèle réussit donc ce test.

Ensuite, nous testons la normalité des résidus en se basant sur un simple test visuel et pour cela on va utiliser la fonction *ols_plot_resid_hist ()* du package *olsrr* dans R. Nous utilisons cette fonction pour tracer un histogramme des résidus (Figure 5).

```
1. library (olsrr)
2. ols_plot_resid_hist (health_mod2)
```

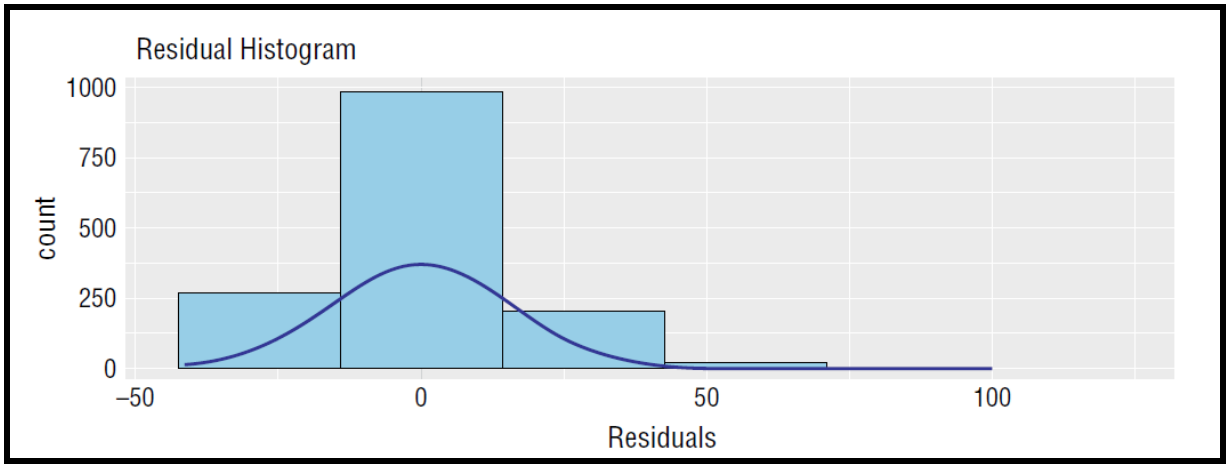

FIGURE 5 : HISTOGRAMME DES RESIDUS

Le graphique ci-dessus est normalement distribué avec une légère inclinaison vers la droite. C'est assez proche d'une distribution normale pour satisfaire notre test.

Ensuite, nous testons la présence d'hétéroscédasticité dans nos résidus (Figure 6). La fonction *ols_plot_resid_fit ()* dans *olsrr* nous permet de visualiser la distribution des valeurs résiduelles en fonction des valeurs ajustées afin de vérifier l'hétéroscédasticité.

```
1. ols_plot_resid_fit (health_mod2)
```

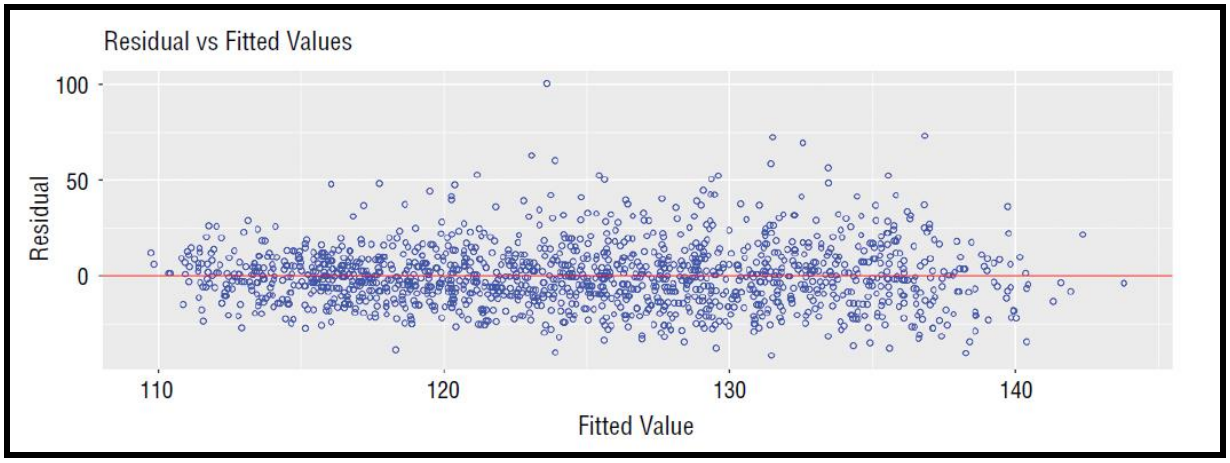

FIGURE 6 : NUAGE DE POINTS DES RESIDUS

Le graphique montre une distribution uniforme des points autour de la ligne d'origine. Il n'y a pas d'hétéroscédasticité dans la distribution des résidus par rapport aux valeurs ajustées.

Ensuite, nous exécutons un test d'autocorrélation résiduelle, le test d'autocorrélation résiduelle le plus populaire est le test Durbin-Watson (DW).

La statistique du test DW varie de 0 à 4, avec des valeurs comprises entre 0 et 2 indiquant une autocorrélation positive, 2 indiquant une autocorrélation nulle et des valeurs comprises entre 2 et 4 indiquant une autocorrélation négative. La fonction

durbinWatsonTest () du package *car* nous fournit un moyen pratique d'obtenir les statistiques de test DW.

```
1. library (car)
2. durbinWatsonTest (health_mod2)
```

```
lag Autocorrelation D-W Statistic p-value
  1     -0.01985291      2.038055   0.456
Alternative hypothesis: rho != 0
```

Avec une statistique Durbin-Watson de 2,04 et une p-value supérieure à 0,05, nous ne pouvons pas rejeter l'hypothèse nulle selon laquelle « aucune autocorrélation de premier ordre n'existe ». Par conséquent, nous pouvons dire que les résidus ne sont pas autocorrélés donc ils sont indépendants.

Le prochain test de diagnostic que nous exécutons est une vérification des points aberrants dans nos données en générant un graphique de la fonction de distance de Cook pour l'ensemble de données (Figure 7). Pour identifier les points influents dans nos données, en fonction de la distance de Cook, nous utiliserons la fonction *ols_plot_cooksd_chart ()* du package *olsrr*.

```
1. ols_plot_cooksd_chart (health_mod2)
```

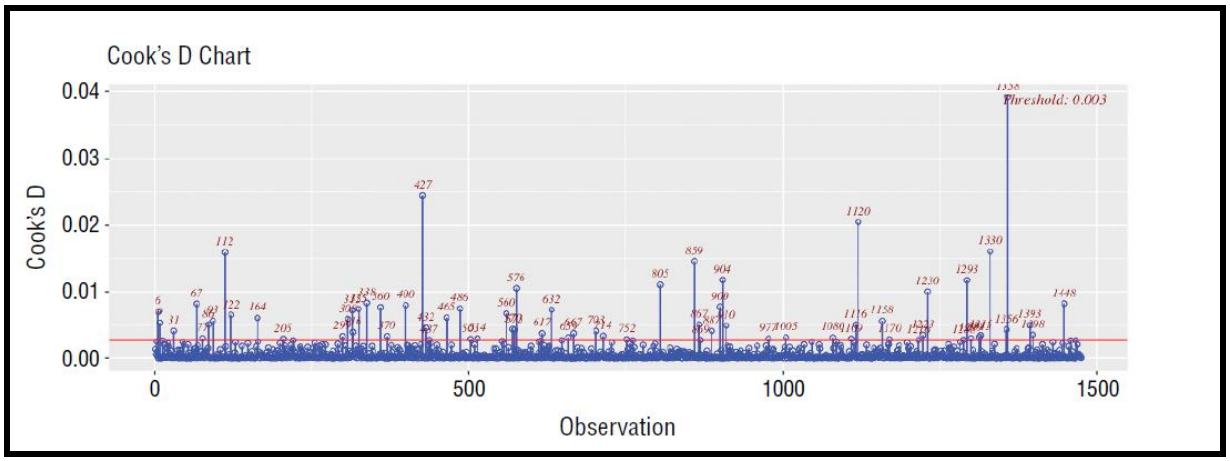

FIGURE 7 : GRAPHIQUE DES DISTANCES DE COOK POUR L'ENSEMBLE DE DONNEES

Le graphique montre qu'il y a en effet plusieurs points aberrants dans nos données. L'observation 1358 se démarque des autres. Jetons un œil aux valeurs observées pour cette observation:

```
1. health[1358,]
```

```
# A tibble: 1 x 9
  systolic weight height   bmi waist   age diabetes smoker fastfood
     <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <fct>    <fct>     <dbl>
1      184   146.   180.  44.9  140.    26 0         0           14
```

On compare ces valeurs avec le résumé statistique de l'ensemble des données, illustré ici:

```
summary(health)
```

```
   systolic         weight          height           bmi             waist      
Min.   : 80.0   Min.   : 29.10   Min.   :141.2   Min.   :13.40   Min.   : 56.2  
1st Qu.:114.0   1st Qu.: 69.15   1st Qu.:163.8   1st Qu.:24.10   1st Qu.: 88.4  
Median :122.0   Median : 81.00   Median :170.3   Median :27.90   Median : 98.9  
Mean   :124.7   Mean   : 83.56   Mean   :170.2   Mean   :28.79   Mean   :100.0  
3rd Qu.:134.0   3rd Qu.: 94.50   3rd Qu.:176.8   3rd Qu.:32.10   3rd Qu.:109.5  
Max.   :224.0   Max.   :203.50   Max.   :200.4   Max.   :62.00   Max.   :176.0  

      age        diabetes  smoker     fastfood      
Min.   :20.00   0:1265    0:770   Min.   : 0.00  
1st Qu.:34.00   1: 210    1:705   1st Qu.: 0.00  
Median :49.00                     Median : 1.00  
Mean   :48.89                     Mean   : 2.14  
3rd Qu.:62.00                     3rd Qu.: 3.00  
Max.   :80.00                     Max.   :22.00  
```

Nous pouvons voir que les valeurs de poids, bmi, de taille, d'âge et de fastfood sont significativement différentes pour l'observation 1358 par rapport à la moyenne et la médiane de ces variables dans l'ensemble de données.

Examinons également la distribution statistique du reste des valeurs aberrantes et comparons celles-ci à la distribution statistique des données sans les valeurs aberrantes. Pour ce faire, nous aurons besoin d'une liste de toutes les observations qui composent nos points influents. Nous devons d'abord obtenir une liste des valeurs d'index pour ces observations. Cela se fait en se référant à la colonne d'observation de l'attribut aberrant de la fonction de distance de Cook.

```
outlier_index <- as.numeric (unlist (
          ols_plot_cooksd_chart (health_mod2)$outliers[,"observation"]))
outlier_index
```

```
 [1]    6    9   31   67   77   86   93  112  122  164  205  299  308  315  316  325
[17]  338  360  370  400  427  432  437  465  486  503  514  560  570  573  576  617
[33]  632  659  667  703  714  752  805  859  867  869  887  900  904  910  977 1005
[49] 1080 1109 1116 1120 1158 1170 1216 1223 1230 1288 1293 1299 1313 1315 1330 1356
[65] 1358 1393 1398 1448
```

Il y a 68 observations dans la liste. Maintenant que nous avons les valeurs des points aberrantes, nous utilisons la commande *summary ()* pour comparer les deux ensembles de données. Tout d'abord, examinons un résumé statistique des seuls points aberrants:

```
summary (health[outlier_index,])
```

```
   systolic         weight           height           bmi             waist
Min.   : 86.0   Min.   : 29.10   Min.   :144.2   Min.   :13.40   Min.   : 56.20
1st Qu.:109.0   1st Qu.: 68.92   1st Qu.:159.5   1st Qu.:23.60   1st Qu.: 92.35
Median :163.0   Median : 82.20   Median :167.2   Median :32.00   Median :111.20
Mean   :149.4   Mean   : 91.73   Mean   :167.2   Mean   :32.26   Mean   :109.81
3rd Qu.:174.0   3rd Qu.:109.03   3rd Qu.:174.2   3rd Qu.:38.42   3rd Qu.:124.92
Max.   :224.0   Max.   :203.50   Max.   :193.3   Max.   :62.00   Max.   :172.20

     age        diabetes smoker    fastfood
Min.   :21.00   0:44     0:29   Min.   : 0.000
1st Qu.:41.75   1:24     1:39   1st Qu.: 0.000
Median :56.00                   Median : 1.000
Mean   :55.50                   Mean   : 2.897
3rd Qu.:68.00                   3rd Qu.: 3.000
Max.   :80.00                   Max.   :18.000
```

Ensuite, comparons cela à un résumé des points de l'ensemble de données à l'exclusion des valeurs aberrantes.

```
1. summary (health[-outlier_index,])
```

```
   systolic         weight           height           bmi             waist
Min.   : 80.0   Min.   : 41.10   Min.   :141.2   Min.   :16.00   Min.   : 65.60
1st Qu.:114.0   1st Qu.: 69.15   1st Qu.:164.0   1st Qu.:24.10   1st Qu.: 88.15
Median :122.0   Median : 81.00   Median :170.4   Median :27.80   Median : 98.50
Mean   :123.5   Mean   : 83.17   Mean   :170.3   Mean   :28.63   Mean   : 99.56
3rd Qu.:134.0   3rd Qu.: 94.10   3rd Qu.:176.8   3rd Qu.:31.90   3rd Qu.:108.80
Max.   :182.0   Max.   :180.20   Max.   :200.4   Max.   :59.00   Max.   :176.00

     age        diabetes smoker     fastfood
Min.   :20.00   0:1221   0:741   Min.   : 0.000
1st Qu.:34.00   1: 186   1:666   1st Qu.: 0.000
Median :48.00                   Median : 1.000
Mean   :48.57                   Mean   : 2.103
3rd Qu.:62.00                   3rd Qu.: 3.000
Max.   :80.00                   Max.   :22.000
```

Nous pouvons voir une différence légère à modérée dans la moyenne et la médiane entre chacune des paires de variables. Alors que les valeurs minimale et maximale pour la plupart des paires sont similaires, nous constatons une différence significative avec les valeurs minimale et maximale de la variable de poids.

Pour améliorer notre modèle, nous devons supprimer ces points influents de notre ensemble de données. Cependant, pour que nous puissions nous référer aux données d'origine, créons une nouvelle version de notre ensemble de données à partir de l'original sans valeurs aberrantes. Nous appelons ce nouvel ensemble de données **health2**.

```
1. health2 <- health[-outlier_index,]
```

Le test de diagnostic final que nous exécutons est le test de multicolinéarité, et pour cela on va calculer le VIF de nos variables grâce à la *fonction ols_vif_tol ()* de *olsrr*.

```
1. ols_vif_tol (health_mod2)
```

```
# A tibble: 8 x 3
  Variables Tolerance   VIF
  <chr>         <dbl> <dbl>
1 weight       0.0104 96.1
2 height       0.0522 19.2
3 bmi          0.0125 80.0
4 waist        0.0952 10.5
5 age          0.588   1.70
6 diabetes1    0.887   1.13
7 smoker1      0.840   1.19
8 fastfood     0.896   1.12
```

Avec un VIF bien supérieur à 10 pour le poids, la taille, bmi et la variable 'waist', il est évident que nous avons un problème de multicolinéarité. Cela n'est pas surprenant, étant donné que le bmi est calculée en divisant le poids par le carré de la taille et que le tour de taille est fortement corrélé au poids d'une personne. Pour résoudre ce problème de multicolinéarité, nous devons soit combiner les variables impactées, soit supprimer certaines d'entre elles. Étant donné que le poids a la tolérance la plus faible parmi les quatre prédicteurs, nous choisissons d'abandonner les trois autres et de conserver le poids.

Avec les modifications que nous avons apportées à nos données et les nouvelles informations dont nous disposons sur notre modèle, on va créer un nouveau modèle de régression linéaire multiple.

```
1. health_mod3 <- lm (data=health2, systolic ~ weight+age+diabetes)
2. summary (health_mod3)
```

```
Call:
lm(formula = systolic ~ weight + age + diabetes, data = health2)

Residuals:
    Min      1Q  Median      3Q     Max
-38.825  -9.004  -0.177   8.222  49.679

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 96.62591    1.93014  50.062  < 2e-16 ***
weight       0.09535    0.01870   5.100 3.87e-07 ***
age          0.38372    0.02218  17.297  < 2e-16 ***
diabetes1    2.62446    1.11859   2.346   0.0191 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.59 on 1403 degrees of freedom
Multiple R-squared:  0.2128,  Adjusted R-squared:  0.2111
F-statistic: 126.4 on 3 and 1403 DF,  p-value: < 2.2e-16
```

Tous nos prédicteurs sont significatifs et tous nos diagnostics de modèle montrent une amélioration par rapport au modèle précédent. Notre modèle explique maintenant 21% de la variabilité de la réponse. C'est encore assez faible, alors essayons de voir si nous pouvons encore améliorer notre modèle.

On va tout d'abord exécuter des tests de diagnostic sur le nouveau modèle comme nous l'avons déjà fait pour le précédent, et commencent par le test de la moyenne nulle des résidus.

```
1. # Test de la moyenne nulle des résidus.
2. mean(health_mod3$residuals)
```

```
[1] -4.032061e-16
```

La moyenne résiduelle est très proche de zéro, Alors, le test est vérifié.

Ensuite, nous testons la normalité des résidus (Figure 8).

```
1. ## Normalité des résidus
2. library(olsrr)
3. ols_plot_resid_hist(health_mod3)
```

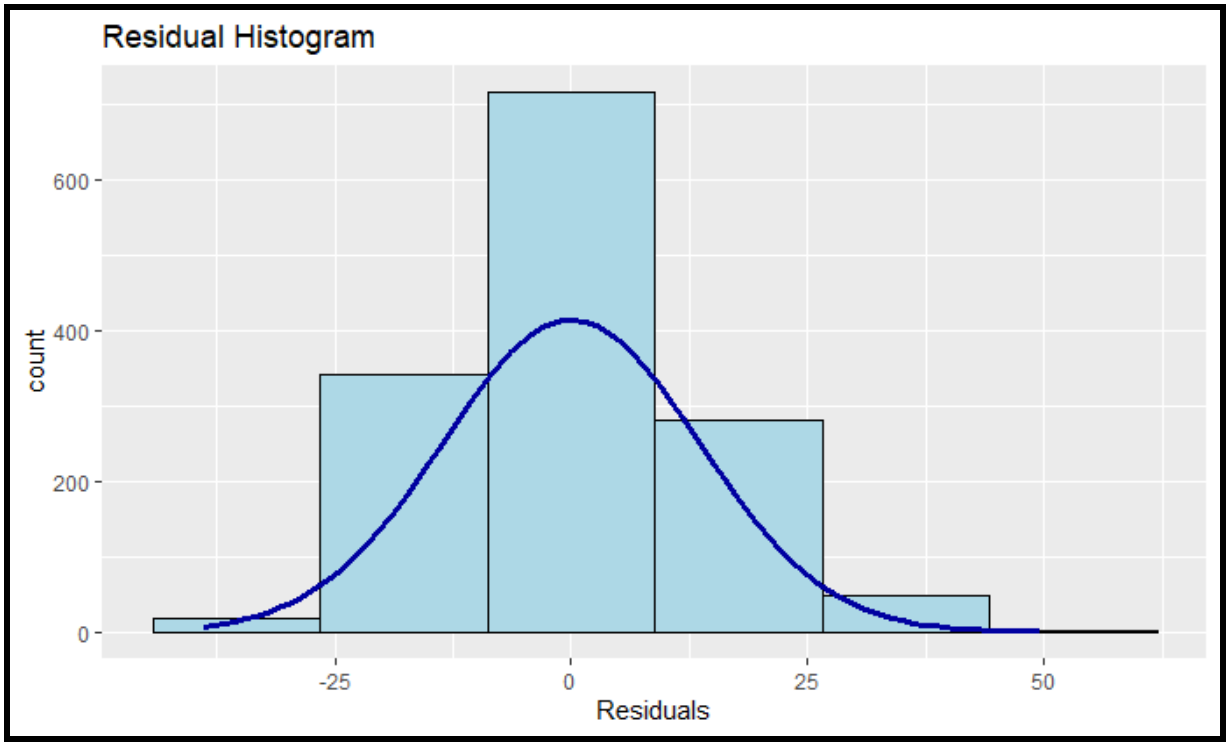

FIGURE 8 : HISTOGRAMME DES RESIDUS POUR LE DEUXIEME MODELE

On peut clairement remarquer qu'on a une distribution normale, alors l'hypothèse de la normalité des résidus est bien vérifiée.

Ensuite, nous testons l'hypothèse d'homoscédasticité (stabilité de la variance) dans nos résidus (Figure 9).

```
1. # Homoscédasticité des résidus
2. ols_plot_resid_fit(health_mod3)
```

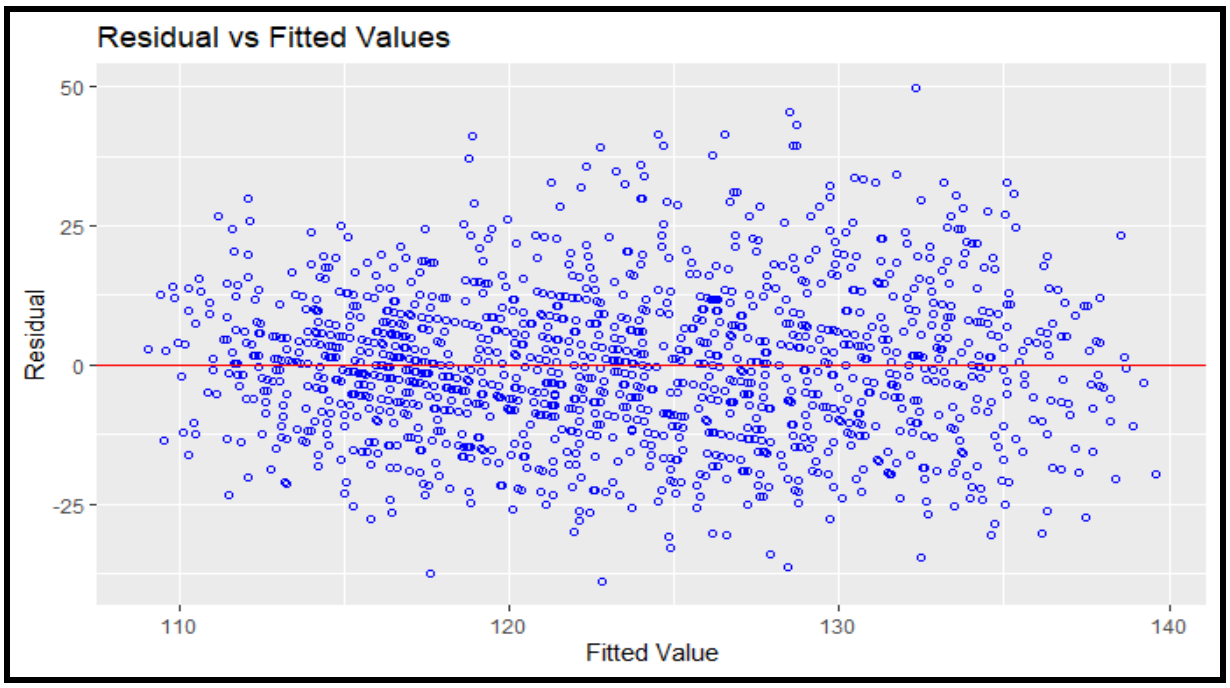

FIGURE 9 : HOMOSCEDASTICITE DES RESIDUS POUR LE DEUXIEME MODELE

On remarque que le nuage des points n'a pas une forme bien spécifique par rapport à la droite d'origine, Alors on peut décider que l'hypothèse d'homoscédasticité est vérifiée c'est-à-dire que la variance est stable.

Ensuite, nous exécutons un test de corrélation des résidus pour vérifier leur indépendance en utilisant le test de durbin-watson.

```
## Autocorrélation résiduelle
## La fonction set.seed () garantit que nous pouvons obtenir la même p-value à chaque fois.
library(car)
set.seed(123)
durbinWatsonTest(health_mod3)
```

```
 lag Autocorrelation D-W Statistic p-value
   1      -0.03548186       2.068932    0.18
 Alternative hypothesis: rho != 0
```

On remarque que la p-value est supérieure à 0,05, alors nous ne pouvons pas rejeter l'hypothèse nulle selon laquelle « aucune autocorrélation de premier ordre n'existe ». Par conséquent, nous pouvons dire que les résidus ne sont pas autocorrélés.

Le prochain test est de vérifier les points aberrants (Figure 10).

```
## Analyse des points aberrants
# Créer le graphique des points aberrants en fonction de la distance de Cook.
ols_plot_cooksd_chart(health_mod3)
```

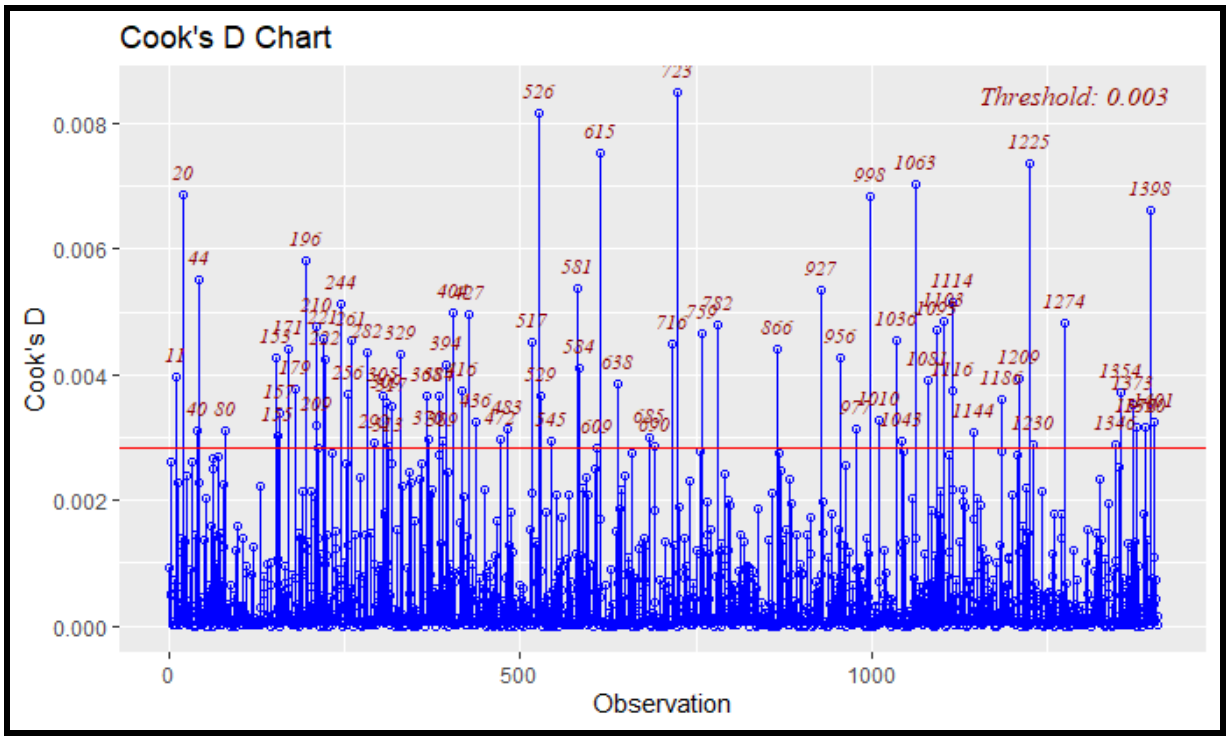

FIGURE 10 : POINTS ABBERANTS POUR LE DEUXIEME MODELE

Le graphique montre qu'il y a en effet des points aberrants dans nos données. Nous allons récupérer la liste de toutes les observations qui composent des points aberrants. Nous devons d'abord obtenir une liste des valeurs d'index pour ces observations. Cela se fait en se référant à la colonne d'observation de l'attribut aberrant de la fonction de distance de Cook.

```
#récupérer les outliers
params1=ols_plot_cooksd_chart(health_mod3)
params2 = params1$plot_env
result1 = params2$f[,"observation"]

# Lister les valeurs aberrantes du graphique ci-dessus.
outlier_index <- as.numeric(unlist(result1))
outlier_index
```

On obtient 78 observations qui composent des points aberrants :

```
[1]   11   20   40   44   80  153  155  157  171  179  196  209  210  221  222
[16]  244  256  261  282  292  305  309  313  317  329  368  370  384  389  394
[31]  404  416  427  436  472  483  517  526  529  545  581  584  609  615  638
[46]  685  690  716  723  759  782  866  927  956  977  998 1010 1036 1043 1063
[61] 1081 1093 1103 1114 1116 1144 1186 1209 1225 1230 1274 1346 1354 1373 1378
[76] 1390 1398 1401
```

Pour améliorer notre modèle, il faut supprimer ces points influents de l'ensemble de données. Comme on a fait auparavant on va créer une nouvelle version de l'ensemble de données à partir de la version précédente sans valeurs aberrantes. Nous appelons ce nouvel ensemble de données **health3**.

```
# Créer un nouvel ensemble de données sans les valeurs aberrantes.
health3 <- health2[-outlier_index,]
```

On va effectuer un dernier test c'est le test de multicolinéarité.

```
## Multicolinéarité
ols_vif_tol(health_mod3)
```

	Variables	Tolerance	VIF
1	weight	0.9814401	1.018911
2	age	0.9301797	1.075061
3	diabetes1	0.9145458	1.093439

On remarque que les valeurs de VIF de l'ensemble des variables ('weight','age','diabetes') sont bien inférieures à 10 alors on peut dire qu'on n'a pas un effet de colinéarité entre les variables.

L'élément suivant que nous considérons est la possibilité d'avoir un effet d'interaction entre les prédicteurs.

Il est raisonnable de s'attendre à ce qu'il puisse y avoir des interactions entre le poids et le diabète et entre l'âge et le diabète, nous allons donc voir est que on a des interactions possibles dans notre modèle.

Possibilité d'avoir des interactions :

```
# interaction entre 'weight' et 'diabetes' ET entre 'age' et 'diabetes'
model_interaction1 = lm(
  data = health3,
  systolic ~ weight * diabetes + age * diabetes
)
summary(model_interaction1)
```

```
Call:
lm(formula = systolic ~ weight * diabetes + age * diabetes, data =
health3)

Residuals:
    Min      1Q  Median      3Q     Max
-37.307  -8.198   0.167   7.819  42.152

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      96.678648   1.896319  50.982  < 2e-16 ***
weight            0.102548   0.018856   5.438  6.4e-08 ***
diabetes1        -1.317173   7.002444  -0.188    0.851
age               0.358798   0.021563  16.640  < 2e-16 ***
weight:diabetes1 -0.001736   0.050934  -0.034    0.973
diabetes1:age     0.081211   0.078488   1.035    0.301
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.15 on 1323 degrees of freedom
Multiple R-squared:  0.2386,    Adjusted R-squared:  0.2357
F-statistic:  82.9 on 5 and 1323 DF,  p-value: < 2.2e-16
```

On peut remarquer que les interactions ont une p-value inferieur a 5% alors on peut décider que l'effet d'interaction entre les variables n'existe pas.

Alors d'après tous ce qu'on a vu précédemment on peut maintenant créer notre modèle final avec les données modifiées et aussi avec les nouvelles informations.

```
health_mod4 <- lm(data=health3, systolic ~ weight+age+diabetes)
summary(health_mod4)
```

```
Call:
lm(formula = systolic ~ weight + age + diabetes, data = health3)

Residuals:
    Min      1Q  Median      3Q     Max
-37.251  -8.343   0.066   7.901  42.118

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 96.48037    1.79195  53.841  < 2e-16 ***
weight       0.10145    0.01749   5.800 8.28e-09 ***
age          0.36503    0.02072  17.619  < 2e-16 ***
diabetes1    3.30617    1.08452   3.048  0.00235 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.15 on 1325 degrees of freedom
Multiple R-squared:  0.2379,     Adjusted R-squared:  0.2362
F-statistic: 137.9 on 3 and 1325 DF,  p-value: < 2.2e-16
```

Tous les paramètres sont significatifs et tous les diagnostics de modèle montrent une amélioration par rapport au modèle précédent. Ce nouveau modèle est plus nuancé, fournissant différentes estimations pour la pression systolique avec des valeurs de poids et d'âge différentes, ainsi que la présence ou non du diabète.

Les résultats peuvent être interprété comme suit :

Avec le diabète :

$$systolic = 96.48 + 0.1 \times (weight) + 0.36 \times (age) + 3.3 \times (diabetes)$$

Sans le diabète :

$$systolic = 96.48 + 0.1 \times (weight) + 0.36 \times (age)$$

IV. CONCLUSION

Comme nous pouvons le voir d'après les résultats précédents, nous avons augmenté le R-carré ajusté à 0,236. Cela signifie que notre modèle explique maintenant 23,6% de la variabilité de la réponse. Il s'agit d'une amélioration significative par rapport à aux modèles précédents.

Le dernier modèle est meilleur que celui avec lequel nous avons commencé mais encore assez bas, ce qui suggère des limites avec les données. Pour obtenir un modèle qui explique mieux la variabilité de notre réponse, nous aurions besoin de plus de prédicteurs en corrélation avec la réponse. Par exemple, nous pourrions vouloir inclure des informations sur le sexe, les antécédents médicaux familiaux et les habitudes d'exercice dans notre modèle.

Cependant, il est également important de noter que lorsque on travaille avec des données comportementales, il est courant de rencontrer des difficultés à construire un modèle qui explique l'essentiel de la variabilité de la réponse. Ceci est dû à la nature imprévisible du comportement humain.