CSCI 6444 : INTRODUCTION TO BIG DATA & ANALYTICS

Class Project 2 – Exploring Variations in Clustering and Predictive Analysis

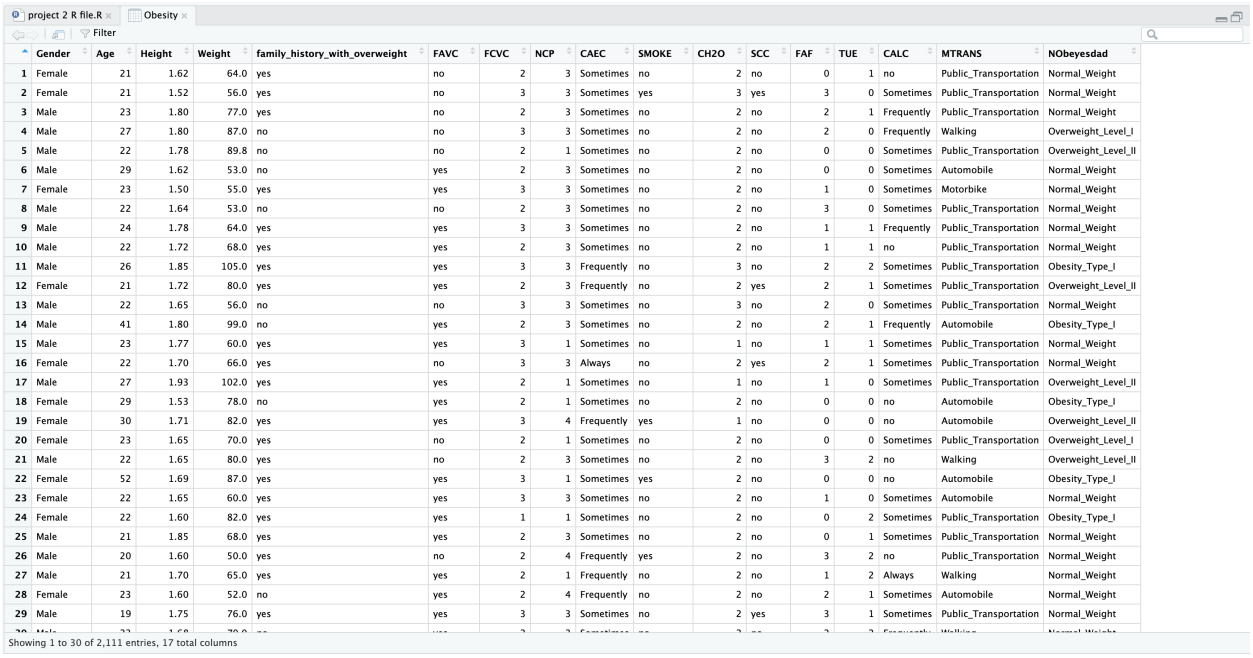
Professor: Stephen Kaisler

Group 20

Yashwanth Raj Varadharajan [G47635180]
Poojasri Mothukuri [G46773587]

Dataset

This data set contains over 2,111 data entries for the people residing from Mexico to Peru to Colombia; accessibility is feasible to check obesity rates in people living in these countries. The study has mentioned different health outcomes via various measures and factors, such as dietary patterns, fitness, and so on, to conclude the overall health of the subject under examination. Each entry is meticulously categorized under the variable 'NObesity' (Obesity Level), which segments the population into seven nuanced groups namely Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II and Obesity Type III.



project 2 R file.R × | Obesity ×

Filter

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	NObeyesdad
1	Female	21	1.62	64.0	yes	no	2	3	Sometimes	no	2	no	0	1	no	Public_Transportation	Normal_Weight
2	Female	21	1.52	56.0	yes	no	3	3	Sometimes	yes	3	yes	3	0	Sometimes	Public_Transportation	Normal_Weight
3	Male	23	1.80	77.0	yes	no	2	3	Sometimes	no	2	no	2	1	Frequently	Public_Transportation	Normal_Weight
4	Male	27	1.80	87.0	no	no	3	3	Sometimes	no	2	no	2	0	Frequently	Walking	Overweight_Level_I
5	Male	22	1.78	89.8	no	no	2	1	Sometimes	no	2	no	0	0	Sometimes	Public_Transportation	Overweight_Level_II
6	Male	29	1.62	53.0	no	yes	2	3	Sometimes	no	2	no	0	0	Sometimes	Automobile	Normal_Weight
7	Female	23	1.50	55.0	yes	yes	3	3	Sometimes	no	2	no	1	0	Sometimes	Motorbike	Normal_Weight
8	Male	22	1.64	53.0	no	no	2	3	Sometimes	no	2	no	3	0	Sometimes	Public_Transportation	Normal_Weight
9	Male	24	1.78	64.0	yes	yes	3	3	Sometimes	no	2	no	1	1	Frequently	Public_Transportation	Normal_Weight
10	Male	22	1.72	68.0	yes	yes	2	3	Sometimes	no	2	no	1	1	no	Public_Transportation	Normal_Weight
11	Male	26	1.85	105.0	yes	yes	3	3	Frequently	no	3	no	2	2	Sometimes	Public_Transportation	Obesity_Type_I
12	Female	21	1.72	80.0	yes	yes	2	3	Frequently	no	2	yes	2	1	Sometimes	Public_Transportation	Overweight_Level_II
13	Male	22	1.65	56.0	no	no	3	3	Sometimes	no	3	no	2	0	Sometimes	Public_Transportation	Normal_Weight
14	Male	41	1.80	99.0	no	yes	2	3	Sometimes	no	2	no	2	1	Frequently	Automobile	Obesity_Type_I
15	Male	23	1.77	60.0	yes	yes	3	1	Sometimes	no	1	no	1	1	Sometimes	Public_Transportation	Normal_Weight
16	Female	22	1.70	66.0	yes	no	3	3	Always	no	2	yes	2	1	Sometimes	Public_Transportation	Normal_Weight
17	Male	27	1.93	102.0	yes	yes	2	1	Sometimes	no	1	no	1	0	Sometimes	Public_Transportation	Overweight_Level_II
18	Female	29	1.53	78.0	no	yes	2	1	Sometimes	no	2	no	0	0	no	Automobile	Obesity_Type_I
19	Female	30	1.71	82.0	yes	yes	3	4	Frequently	yes	1	no	0	0	no	Automobile	Overweight_Level_II
20	Female	23	1.65	70.0	yes	no	2	1	Sometimes	no	2	no	0	0	Sometimes	Public_Transportation	Overweight_Level_I
21	Male	22	1.65	80.0	yes	no	2	3	Sometimes	no	2	no	3	2	no	Walking	Overweight_Level_II
22	Female	52	1.69	87.0	yes	yes	3	1	Sometimes	yes	2	no	0	0	no	Automobile	Obesity_Type_I
23	Female	22	1.65	60.0	yes	yes	3	3	Sometimes	no	2	no	1	0	Sometimes	Automobile	Normal_Weight
24	Female	22	1.60	82.0	yes	yes	1	1	Sometimes	no	2	no	0	2	Sometimes	Public_Transportation	Obesity_Type_I
25	Male	21	1.85	68.0	yes	yes	2	3	Sometimes	no	2	no	0	1	Sometimes	Public_Transportation	Normal_Weight
26	Male	20	1.60	50.0	yes	no	2	4	Frequently	yes	2	no	3	2	no	Public_Transportation	Normal_Weight
27	Male	21	1.70	65.0	yes	yes	2	1	Frequently	no	2	no	1	2	Always	Walking	Normal_Weight
28	Female	23	1.60	52.0	no	yes	2	4	Frequently	no	2	no	2	1	Sometimes	Automobile	Normal_Weight
29	Male	19	1.75	76.0	yes	yes	3	3	Sometimes	no	2	yes	3	1	Sometimes	Public_Transportation	Normal_Weight

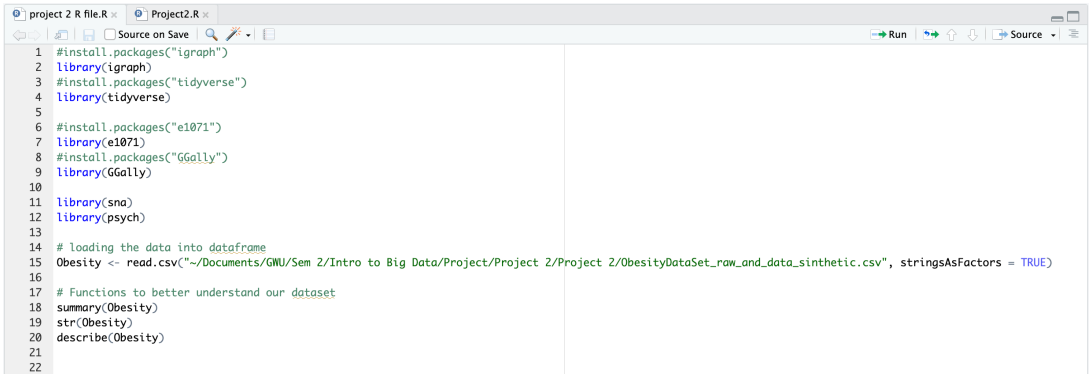
Showing 1 to 30 of 2,111 entries, 17 total columns

Importing the specified dataset

In this step we import the dataset into our program. Only then will we be able to perform the necessary operations. To achieve this goal we use certain functions.

setwds - This function specifies the present working directory from which all files are to be taken. Inside this specified directory is where the dataset is stored.

read.csv – This function reads the data from the csv file and stores it in a variable called Obesity.



```r
#install.packages("igraph")
library(igraph)
#install.packages("tidyverse")
library(tidyverse)

#install.packages("e1071")
library(e1071)
#install.packages("GGally")
library(GGally)

library(sna)
library(psych)

# loading the data into dataframe
Obesity <- read.csv("~/Documents/GWU/Sem 2/Intro to Big Data/Project/Project 2/Project 2/ObesityDataSet_raw_and_data_sinthetic.csv", stringsAsFactors = TRUE)

# Functions to better understand our dataset
summary(Obesity)
str(Obesity)
describe(Obesity)
```

Dataset Information

We are going to use three functions namely summary, str and describe to better understand the dataset.

summary - The summary function provides a summary of the statistical properties of numeric variables in a data frame or matrix. It calculates various summary statistics like minimum, maximum, median, mean, and quartiles for each numeric variable.

str - The str() function (short for "structure") provides a compact display of the internal structure of R objects. It gives you a concise summary of the structure of an R object, showing the data type and length of each component.

describe – The describe function provides a summary of descriptive statistics for variables in a dataset. It gives insights into the central tendency, dispersion, and shape of the data distribution, including mean, median, standard deviation, minimum, maximum, and quartiles.

```
> summary(Obesity)
    Gender          Age            Height          Weight       family_history_with_overweight  FAVC
 Female:1043   Min.   :14.00   Min.   :1.450   Min.   : 39.00   no : 385                       no : 245
 Male  :1068   1st Qu.:19.95   1st Qu.:1.630   1st Qu.: 65.47   yes:1726                       yes:1866
               Median :22.78   Median :1.700   Median : 83.00
               Mean   :24.31   Mean   :1.702   Mean   : 86.59
               3rd Qu.:26.00   3rd Qu.:1.768   3rd Qu.:107.43
               Max.   :61.00   Max.   :1.980   Max.   :173.00

      FCVC            NCP                CAEC      SMOKE           CH2O        SCC            FAF
 Min.   :1.000   Min.   :1.000   Always    :  53   no :2067   Min.   :1.000   no :2015   Min.   :0.0000
 1st Qu.:2.000   1st Qu.:2.659   Frequently: 242   yes:  44   1st Qu.:1.585   yes:  96   1st Qu.:0.1245
 Median :2.386   Median :3.000   no        :  51              Median :2.000              Median :1.0000
 Mean   :2.419   Mean   :2.686   Sometimes :1765              Mean   :2.008              Mean   :1.0103
 3rd Qu.:3.000   3rd Qu.:3.000                                3rd Qu.:2.477              3rd Qu.:1.6667
 Max.   :3.000   Max.   :4.000                                Max.   :3.000              Max.   :3.0000

      TUE               CALC                        MTRANS                 NObeyesdad
 Min.   :0.0000   Always    :   1   Automobile           : 457   Insufficient_Weight:272
 1st Qu.:0.0000   Frequently:  70   Bike                 :   7   Normal_Weight      :287
 Median :0.6253   no        : 639   Motorbike            :  11   Obesity_Type_I     :351
 Mean   :0.6579   Sometimes :1401   Public_Transportation:1580   Obesity_Type_II    :297
 3rd Qu.:1.0000                     Walking              :  56   Obesity_Type_III   :324
 Max.   :2.0000                                                 Overweight_Level_I :290
                                                                Overweight_Level_II:290

>
```

```
> str(Obesity)
'data.frame':   2111 obs. of  17 variables:
 $ Gender                        : Factor w/ 2 levels "Female","Male": 1 1 2 2 2 2 1 2 2 2 ...
 $ Age                           : num  21 21 23 27 22 29 23 22 24 22 ...
 $ Height                        : num  1.62 1.52 1.8 1.8 1.78 1.62 1.5 1.64 1.78 1.72 ...
 $ Weight                        : num  64 56 77 87 89.8 53 55 53 64 68 ...
 $ family_history_with_overweight: Factor w/ 2 levels "no","yes": 2 2 2 1 1 1 2 1 2 2 ...
 $ FAVC                          : Factor w/ 2 levels "no","yes": 1 1 1 1 1 2 2 1 2 2 ...
 $ FCVC                          : num  2 3 2 3 2 2 3 2 3 2 ...
 $ NCP                           : num  3 3 3 3 1 3 3 3 3 3 ...
 $ CAEC                          : Factor w/ 4 levels "Always","Frequently",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ SMOKE                         : Factor w/ 2 levels "no","yes": 1 2 1 1 1 1 1 1 1 1 ...
 $ CH2O                          : num  2 3 2 2 2 2 2 2 2 2 ...
 $ SCC                           : Factor w/ 2 levels "no","yes": 1 2 1 1 1 1 1 1 1 1 ...
 $ FAF                           : num  0 3 2 2 0 0 1 3 1 1 ...
 $ TUE                           : num  1 0 1 0 0 0 0 0 1 1 ...
 $ CALC                          : Factor w/ 4 levels "Always","Frequently",..: 3 4 2 2 4 4 4 4 2 3 ...
 $ MTRANS                        : Factor w/ 5 levels "Automobile","Bike",..: 4 4 4 5 4 1 3 4 4 4 ...
 $ NObeyesdad                    : Factor w/ 7 levels "Insufficient_Weight",..: 2 2 2 6 7 2 2 2 2 2 ...
>
```

```
> describe(Obesity)
                               vars    n  mean    sd median trimmed   mad   min    max  range  skew
Gender*                           1 2111  1.51  0.50   2.00    1.51  0.00  1.00   2.00   1.00 -0.02
Age                               2 2111 24.31  6.35  22.78   23.34  4.78 14.00  61.00  47.00  1.53
Height                            3 2111  1.70  0.09   1.70    1.70  0.10  1.45   1.98   0.53 -0.01
Weight                            4 2111 86.59 26.19  83.00   85.82 32.22 39.00 173.00 134.00  0.26
family_history_with_overweight*   5 2111  1.82  0.39   2.00    1.90  0.00  1.00   2.00   1.00 -1.64
FAVC*                             6 2111  1.88  0.32   2.00    1.98  0.00  1.00   2.00   1.00 -2.40
FCVC                              7 2111  2.42  0.53   2.39    2.46  0.57  1.00   3.00   2.00 -0.43
NCP                               8 2111  2.69  0.78   3.00    2.77  0.00  1.00   4.00   3.00 -1.11
CAEC*                             9 2111  3.67  0.78   4.00    3.87  0.00  1.00   4.00   3.00 -2.13
SMOKE*                           10 2111  1.02  0.14   1.00    1.00  0.00  1.00   2.00   1.00  6.70
CH2O                             11 2111  2.01  0.61   2.00    2.01  0.67  1.00   3.00   2.00 -0.10
SCC*                             12 2111  1.05  0.21   1.00    1.00  0.00  1.00   2.00   1.00  4.36
FAF                              13 2111  1.01  0.85   1.00    0.94  1.19  0.00   3.00   3.00  0.50
TUE                              14 2111  0.66  0.61   0.63    0.59  0.72  0.00   2.00   2.00  0.62
CALC*                            15 2111  3.63  0.55   4.00    3.70  0.00  1.00   4.00   3.00 -1.17
MTRANS*                          16 2111  3.37  1.26   4.00    3.55  0.00  1.00   5.00   4.00 -1.28
NObeyesdad*                      17 2111  4.02  1.95   4.00    4.02  2.97  1.00   7.00   6.00  0.01
                               kurtosis   se
Gender*                           -2.00 0.01
Age                                2.81 0.14
Height                            -0.57 0.00
Weight                            -0.70 0.57
family_history_with_overweight*    0.70 0.01
FAVC*                              3.74 0.01
FCVC                              -0.64 0.01
NCP                                0.38 0.02
CAEC*                              3.06 0.02
SMOKE*                            42.95 0.00
CH2O                              -0.88 0.01
SCC*                              17.02 0.00
FAF                               -0.62 0.02
TUE                               -0.55 0.01
CALC*                              0.46 0.01
MTRANS*                           -0.20 0.03
NObeyesdad*                       -1.19 0.04
```

Converting Alphanumeric values to Numeric values

We have defined a function that converts categorical variables represented as factors in given data frame to numeric values based on specified levels order. The function takes three arguments: the data frame, the name of the column containing the factor variable, and the desired order of levels. Inside the function, it first converts the specified column to a factor using the provided levels order. Then, it converts this factor to numeric values using as.numeric(). Finally, it replaces the original factor column in the data frame with the newly created numeric column.

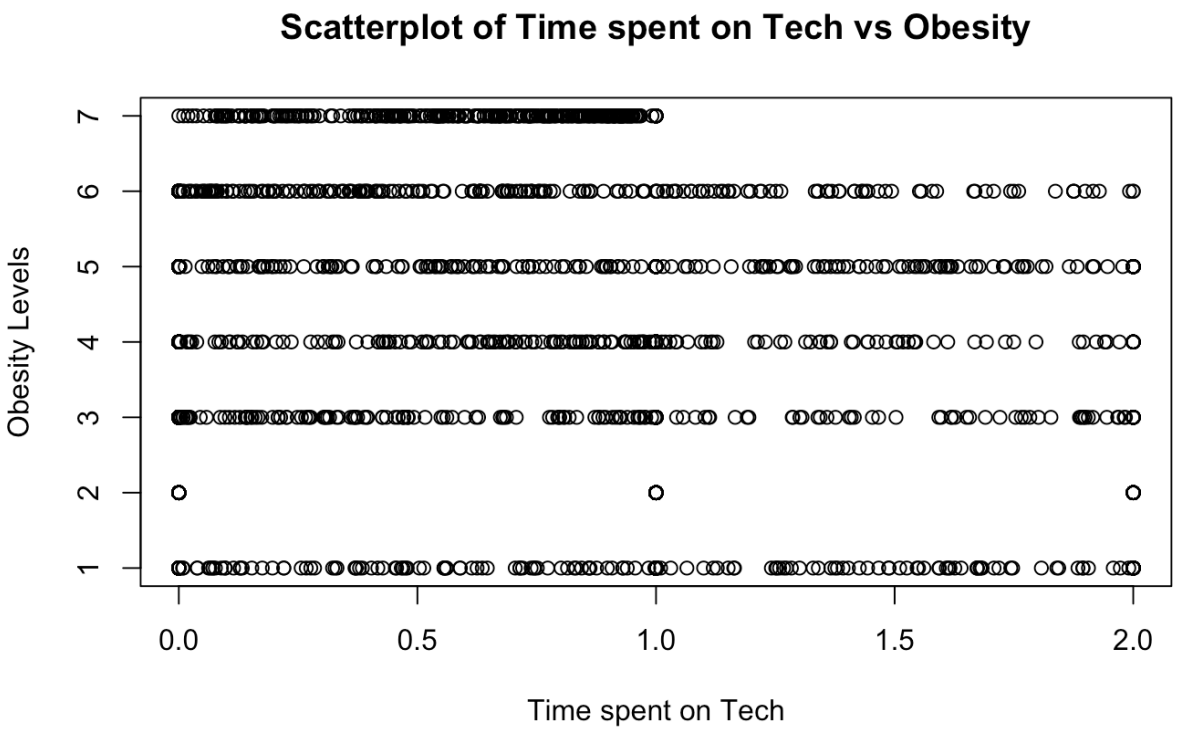
```
> # Creating a function to convert factors to numeric based on their levels
> convert_factor_to_numeric <- function(data_frame, column_name, levels_order) {
+   factor_column <- factor(data_frame[[column_name]], levels = levels_order)
+   numeric_column <- as.numeric(factor_column)
+   data_frame[[column_name]] <- numeric_column
+   return(data_frame)
+ }
> Obesity <- convert_factor_to_numeric(Obesity, "Gender", c("Male", "Female"))
> Obesity <- convert_factor_to_numeric(Obesity, "family_history_with_overweight", c("no", "yes"))
> Obesity <- convert_factor_to_numeric(Obesity, "FAVC", c("no", "yes"))
> Obesity <- convert_factor_to_numeric(Obesity, "CAEC", c("no", "Sometimes", "Frequently", "Always"))
> Obesity <- convert_factor_to_numeric(Obesity, "SMOKE", c("no", "yes"))
> Obesity <- convert_factor_to_numeric(Obesity, "SCC", c("no", "yes"))
> Obesity <- convert_factor_to_numeric(Obesity, "CALC", c("no", "Sometimes", "Frequently", "Always"))
> Obesity <- convert_factor_to_numeric(Obesity, "MTRANS", c("Automobile", "Motorbike", "Bike", "Public_Transportation", "Walking"))
> # converting target variable 'NObeyesdad' to numeric
> Obesity <- convert_factor_to_numeric(Obesity, "NObeyesdad", c("Insufficient_Weight", "Normal_Weight", "Overweight_Level_I", "Overweight_Level_II", "Obesity_Type_I", "O
besity_Type_II", "Obesity_Type_III"))
> str(Obesity)
'data.frame':   2111 obs. of  17 variables:
 $ Gender                        : num  2 2 1 1 1 1 2 1 1 1 ...
 $ Age                           : num  21 21 23 27 22 29 23 22 24 22 ...
 $ Height                        : num  1.62 1.52 1.8 1.8 1.78 1.62 1.5 1.64 1.78 1.72 ...
 $ Weight                        : num  64 56 77 87 89.8 53 55 53 64 68 ...
 $ family_history_with_overweight: num  2 2 2 1 1 1 2 1 2 2 ...
 $ FAVC                          : num  1 1 1 1 1 2 2 1 2 2 ...
 $ FCVC                          : num  2 3 2 3 2 2 3 2 3 2 ...
 $ NCP                           : num  3 3 3 3 1 3 3 3 3 3 ...
 $ CAEC                          : num  2 2 2 2 2 2 2 2 2 2 ...
 $ SMOKE                         : num  1 2 1 1 1 1 1 1 1 1 ...
 $ CH2O                          : num  2 3 2 2 2 2 2 2 2 2 ...
 $ SCC                           : num  1 2 1 1 1 1 1 1 1 1 ...
 $ FAF                           : num  0 3 2 2 0 0 1 3 1 1 ...
 $ TUE                           : num  1 0 1 0 0 0 0 0 1 1 ...
 $ CALC                          : num  1 2 3 3 2 2 2 2 3 1 ...
 $ MTRANS                        : num  4 4 4 5 4 1 2 4 4 4 ...
 $ NObeyesdad                    : num  2 2 2 3 4 2 2 2 2 2 ...
```

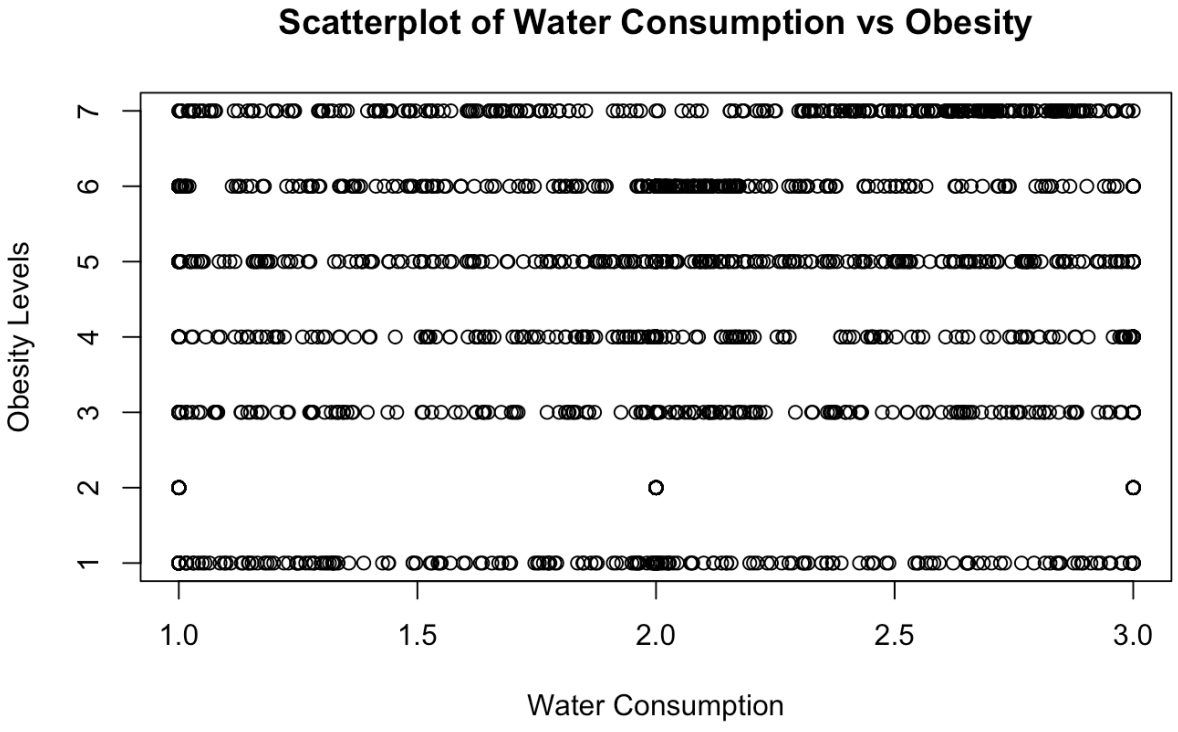
Attributes Analysis

We have defined a function to generate scatterplots of obesity levels against various factors. It takes two arguments: x, representing the variable to be plotted on the x-axis, and y_label, which specifies the label for the y-axis. Inside the function, it utilizes the plot() function to create a scatterplot, with x on the x-axis and Obesity$NObeyesdad (presumably the obesity levels) on the y-axis.

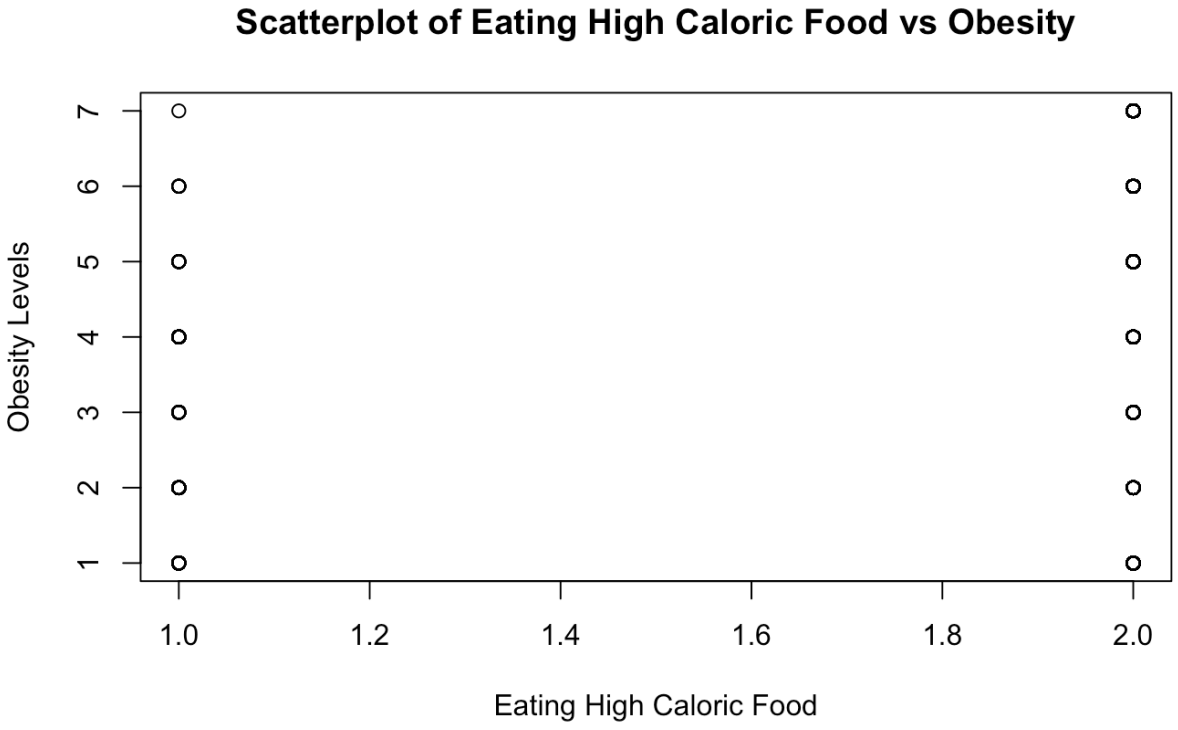
Then we define the cor() function. This function first selects the numeric columns in the dataset using sapply(Obesity, is.numeric) and then calculates the correlation matrix using the selected numeric columns. The option use = "complete.obs" is used to handle missing values by considering complete cases only. The resulting correlation matrix is then printed.

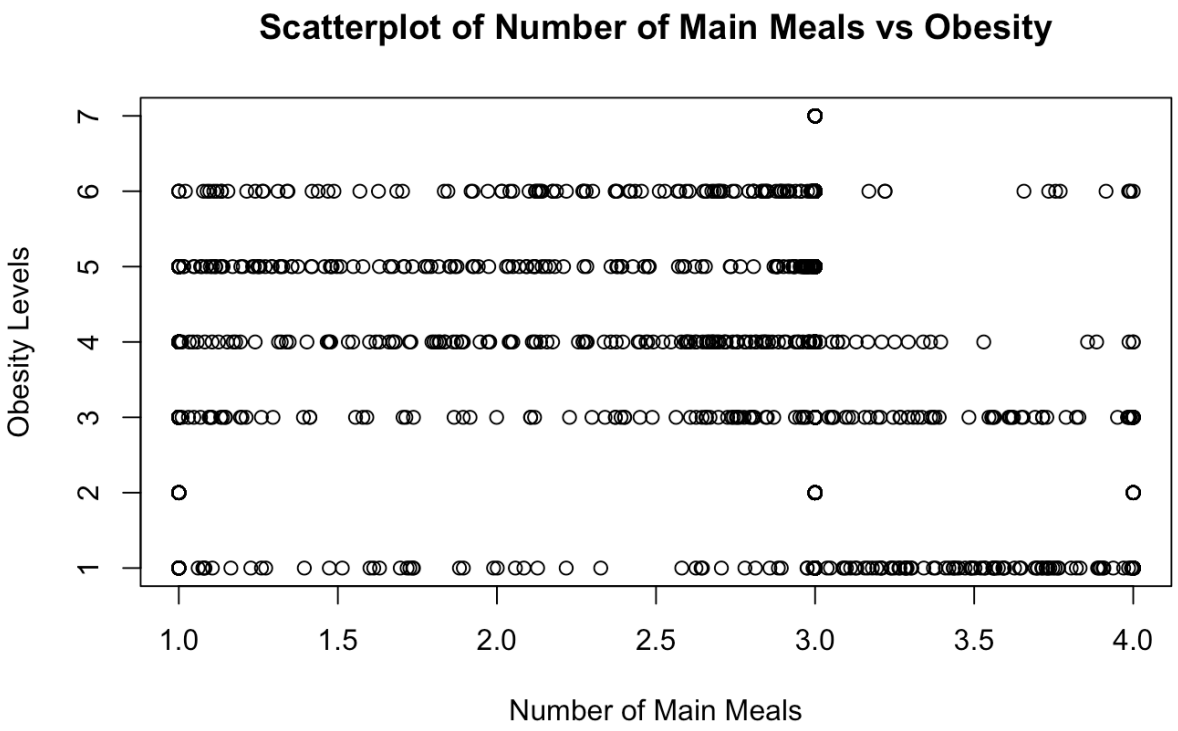
Next, the code checks if the "corrplot" package is installed, and if not, it installs the package using install.packages("corrplot"). Once the package is installed, it loads it into the current R session using library(corrplot). Finally, the corrplot() function from the "corrplot" package is used to visualize the correlation matrix.

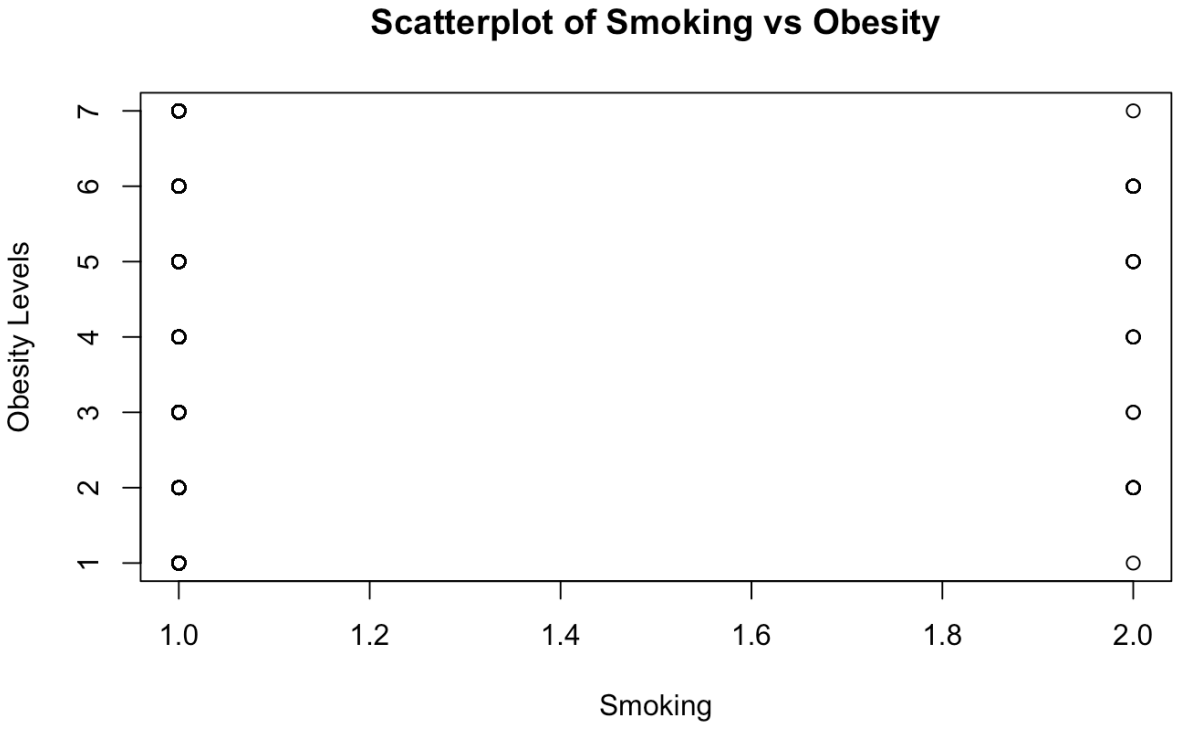

Scatterplot of Time spent on Tech vs Obesity
Obesity Levels
1
2
3
4
5
6
7
0.0
0.5
1.0
1.5
2.0
Time spent on Tech

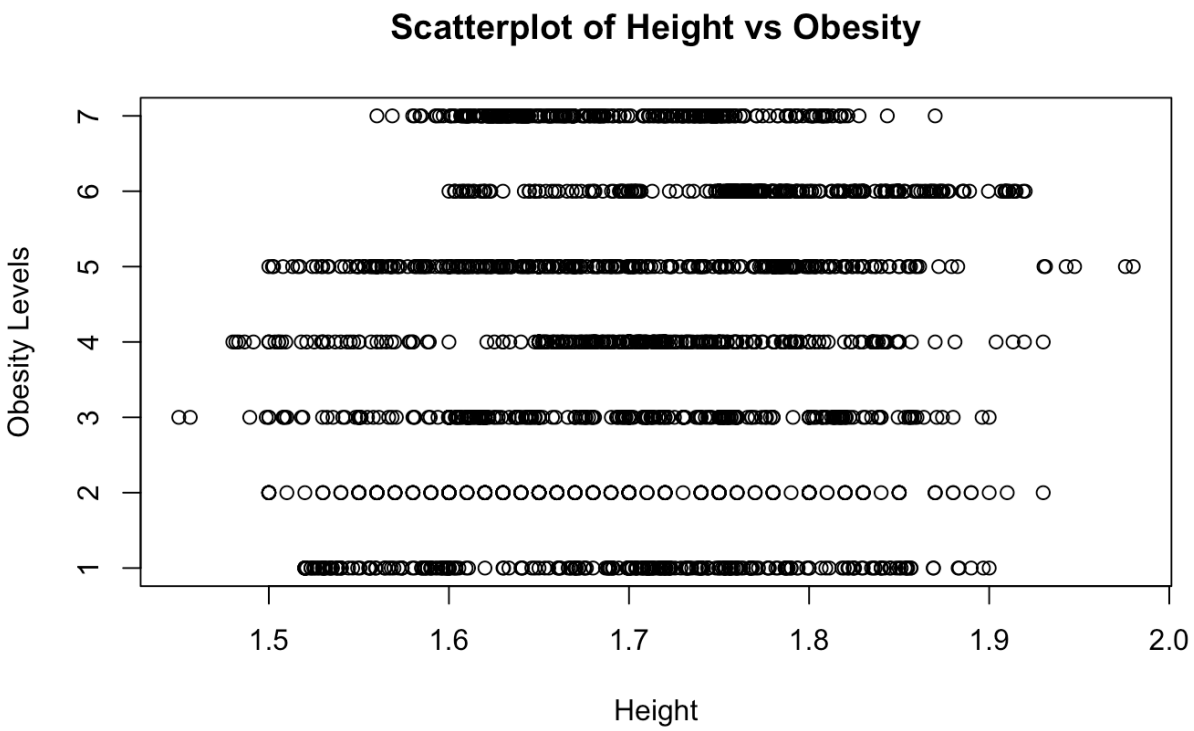

Scatterplot of Water Consumption vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.0
1.5
2.0
2.5
3.0
Water Consumption

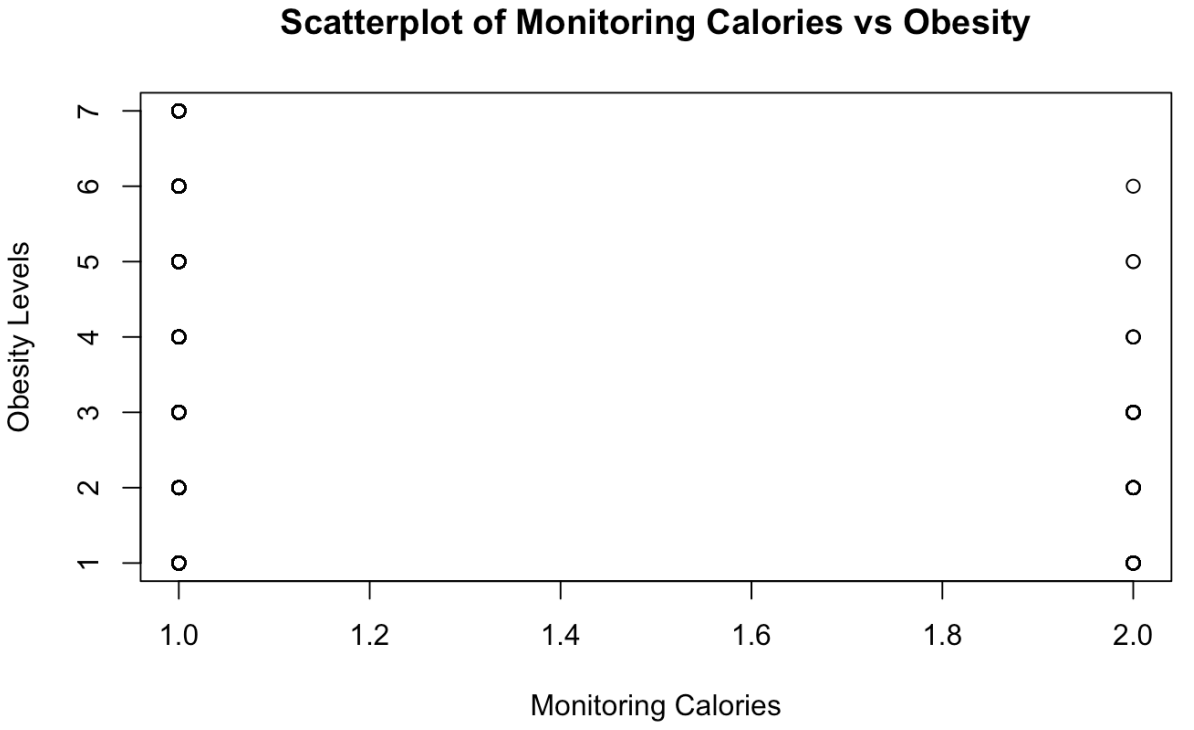

Scatterplot of Eating Vegetables vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.0
1.5
2.0
2.5
3.0
Eating Vegetables

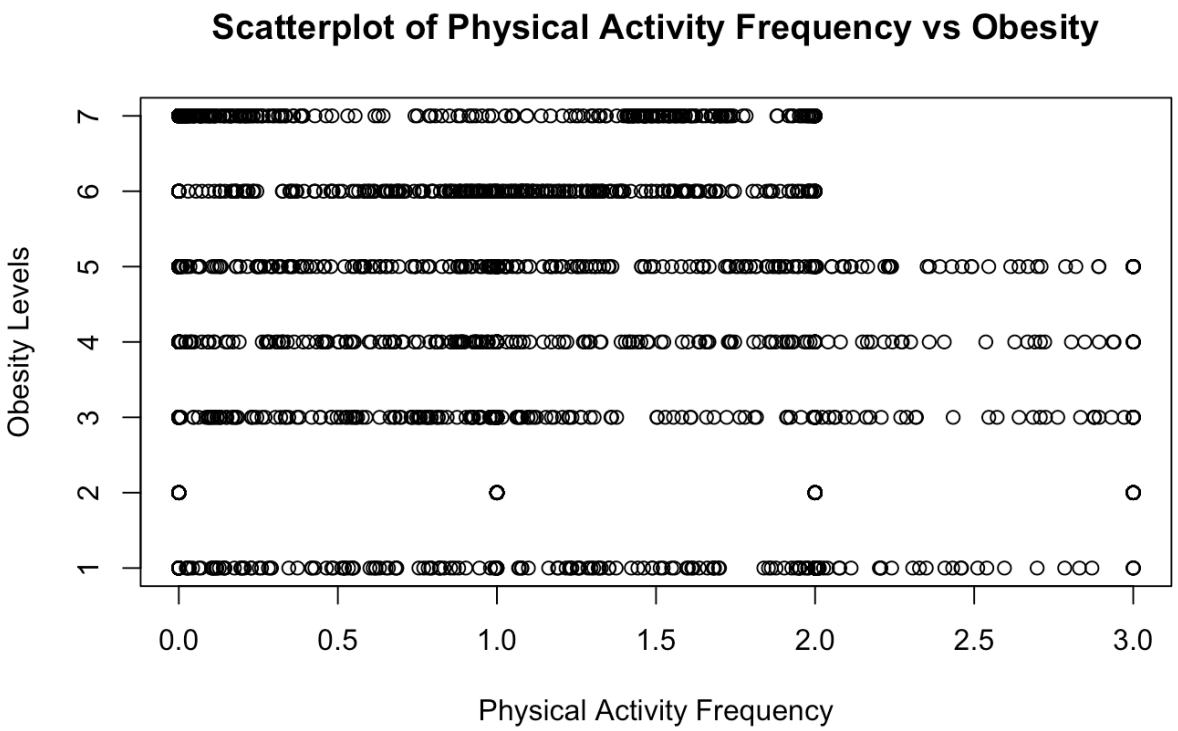

Scatterplot of Eating High Caloric Food vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.0
1.2
1.4
1.6
1.8
2.0
Eating High Caloric Food

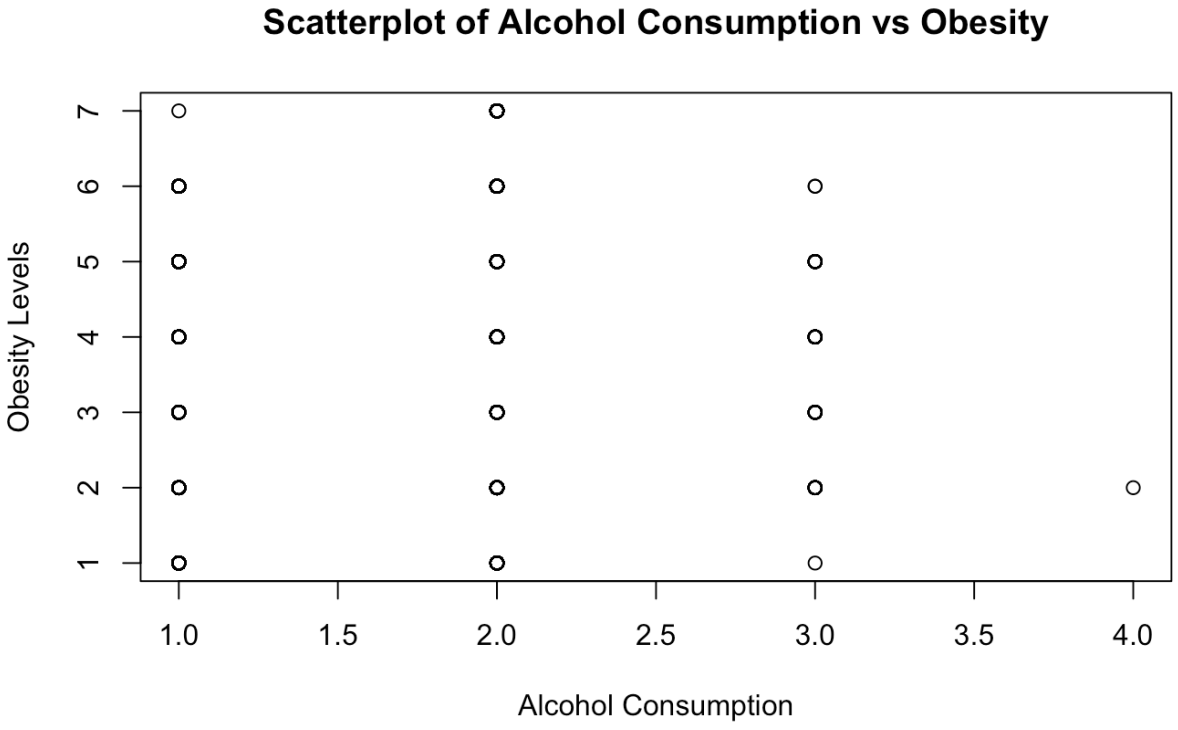

Scatterplot of Number of Main Meals vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Number of Main Meals

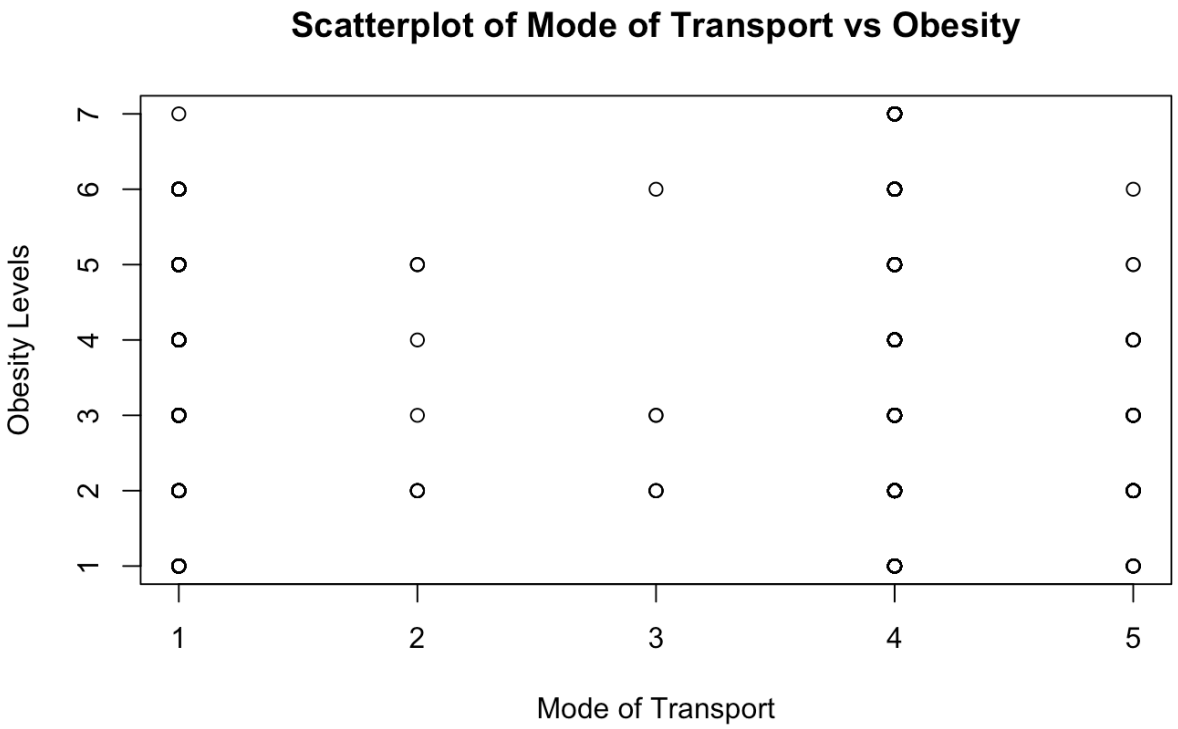

Scatterplot of Smoking vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.0
1.2
1.4
1.6
1.8
2.0
Smoking

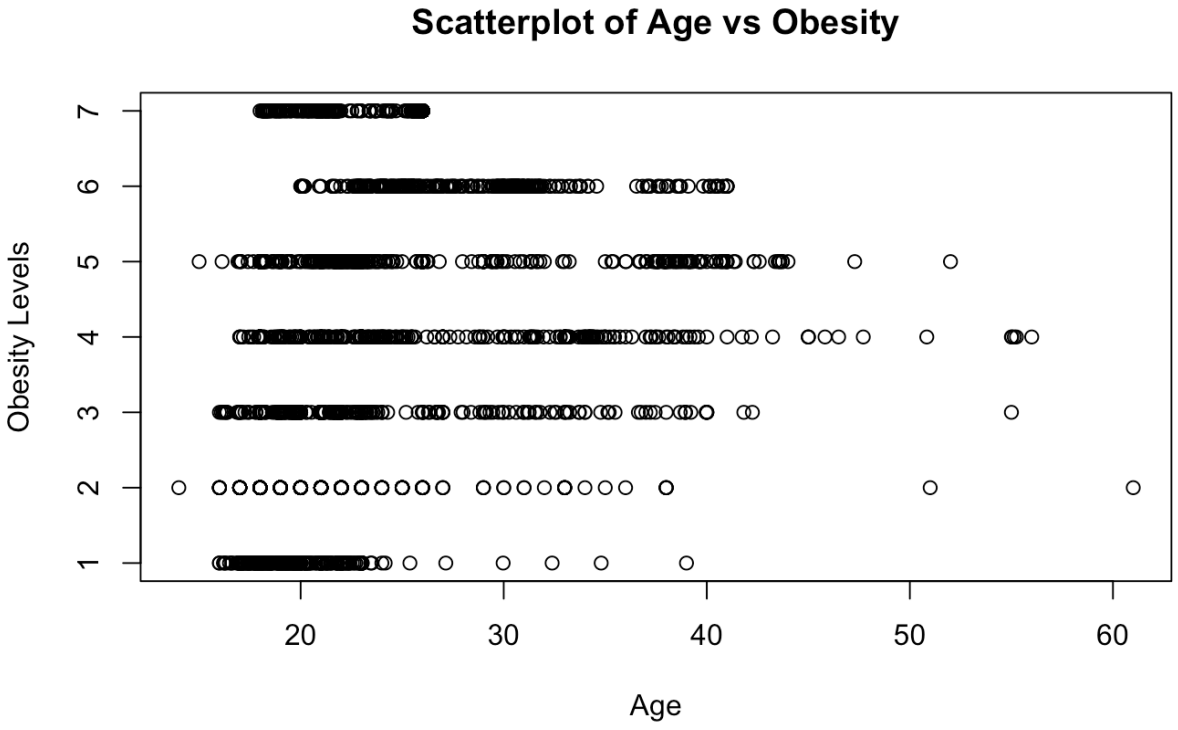

Scatterplot of Height vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.5
1.6
1.7
1.8
1.9
2.0
Height


Scatterplot of Monitoring Calories vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.0
1.2
1.4
1.6
1.8
2.0
Monitoring Calories

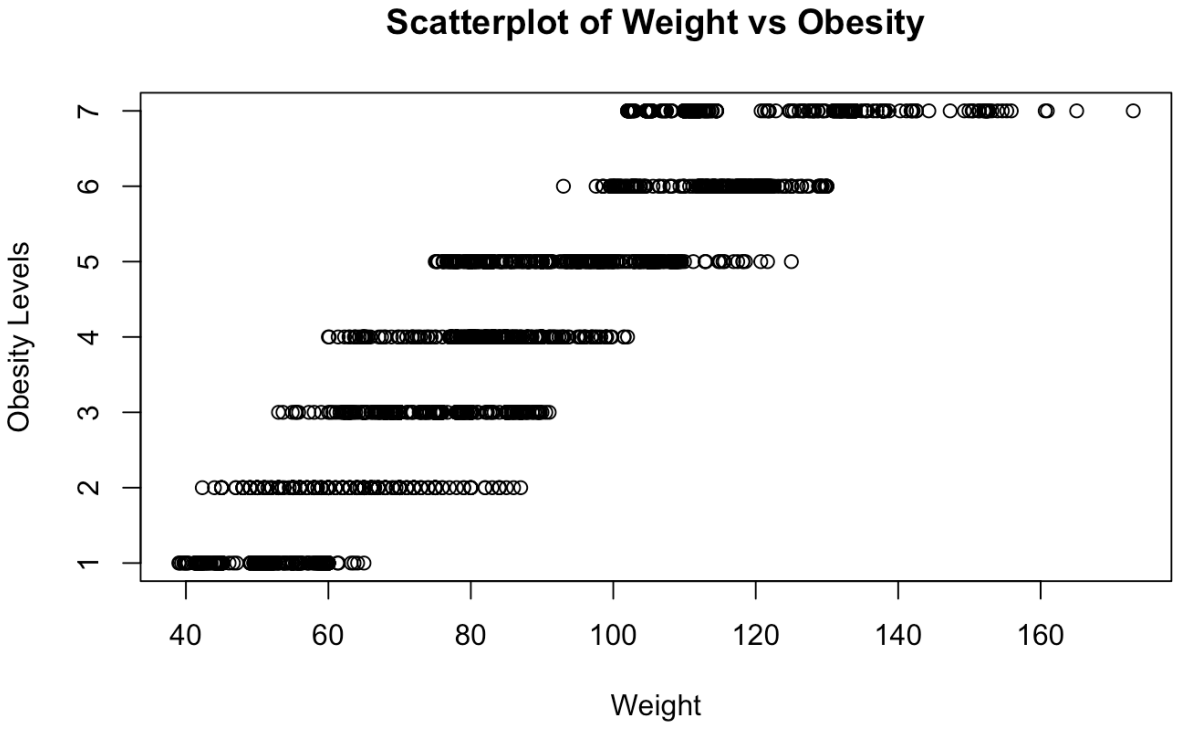

Scatterplot of Physical Activity Frequency vs Obesity
Obesity Levels
1
2
3
4
5
6
7
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Physical Activity Frequency

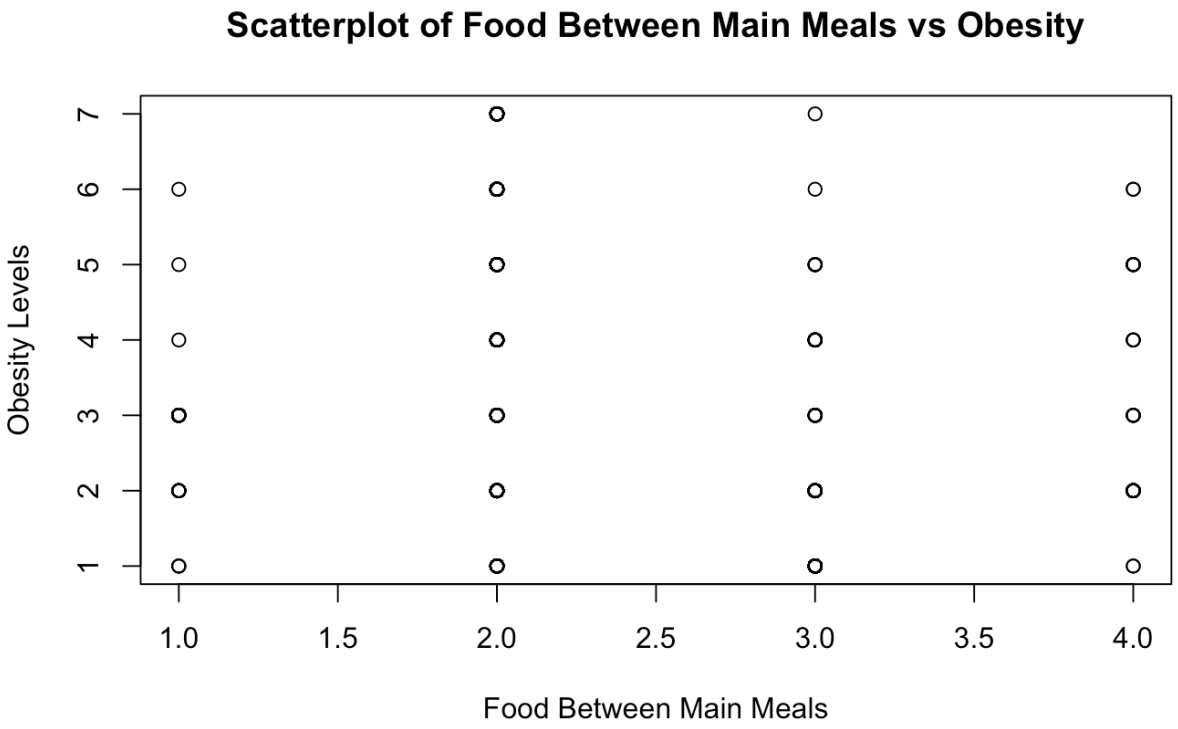

Scatterplot of Alcohol Consumption vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Alcohol Consumption

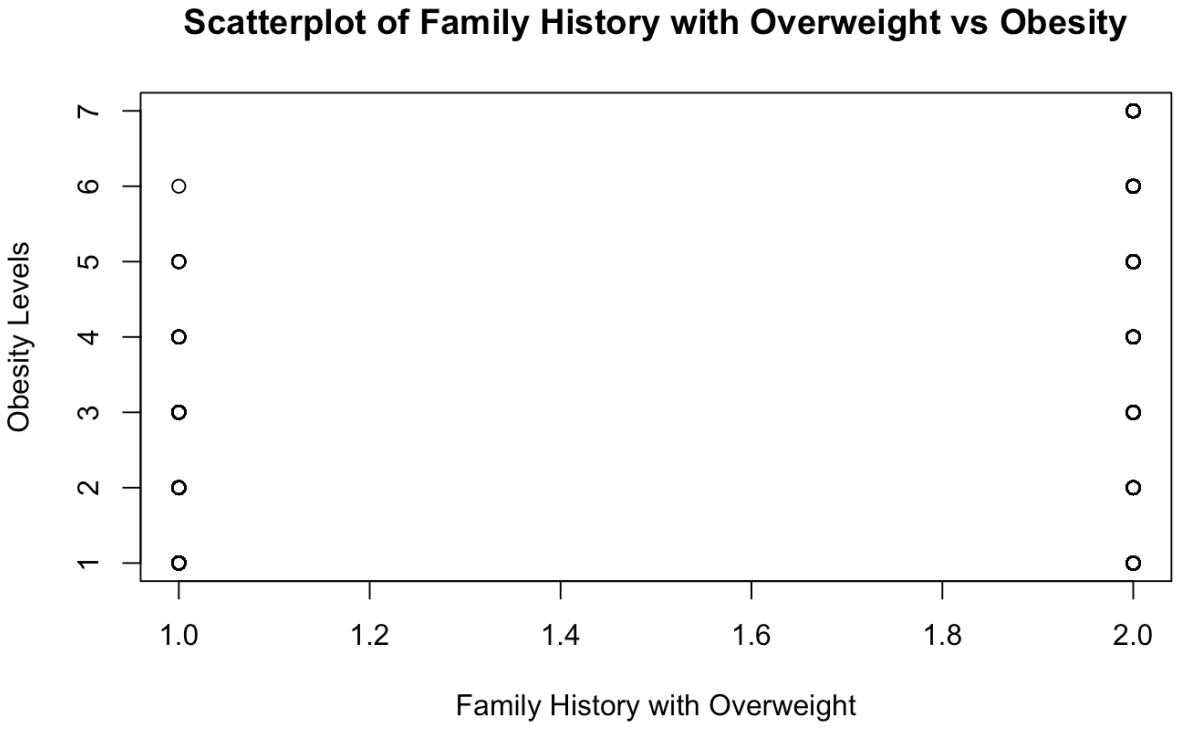

Scatterplot of Mode of Transport vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1
2
3
4
5
Mode of Transport


Scatterplot of Age vs Obesity
Obesity Levels
1
2
3
4
5
6
7
20
30
40
50
60
Age

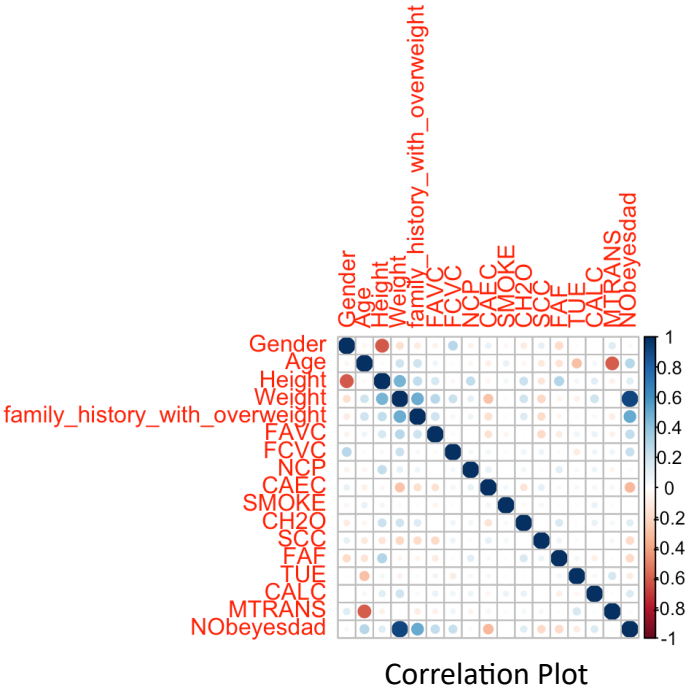

Scatterplot of Gender vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.0
1.2
1.4
1.6
1.8
2.0
Gender


Scatterplot of Weight vs Obesity
Obesity Levels
1
2
3
4
5
6
7
40
60
80
100
120
140
160
Weight


Scatterplot of Food Between Main Meals vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Food Between Main Meals


Scatterplot of Family History with Overweight vs Obesity
Obesity Levels
1
2
3
4
5
6
7
1.0
1.2
1.4
1.6
1.8
2.0
Family History with Overweight

```
> print(cor_matrix)
                                      Gender         Age      Height       Weight
Gender                          1.000000000 -0.04839420 -0.61846630 -0.161667575
Age                            -0.048394197  1.00000000 -0.02595813  0.202560104
Height                         -0.618466297 -0.02595813  1.00000000  0.463136117
Weight                         -0.161667575  0.20256010  0.46313612  1.000000000
family_history_with_overweight -0.102512133  0.20572533  0.24768389  0.496820377
FAVC                           -0.064933774  0.06390169  0.17836378  0.272300490
FCVC                            0.274504782  0.01629089 -0.03812106  0.216124705
NCP                            -0.067599988 -0.04394373  0.24367173  0.107468988
CAEC                            0.091543343 -0.08373870 -0.04881820 -0.287493463
SMOKE                          -0.044698091  0.09198745  0.05549938  0.025746413
CH2O                           -0.107929676 -0.04530386  0.21337592  0.200575387
SCC                             0.102633482 -0.11628285 -0.13375278 -0.201906340
FAF                            -0.189606957 -0.14493833  0.29470900 -0.051436270
TUE                            -0.017269473 -0.29693059  0.05191167 -0.071561359
CALC                            0.007615872  0.04448711  0.12973186  0.206676696
MTRANS                          0.137378117 -0.60231696 -0.07161593  0.005742124
NObeyesdad                      0.031463618  0.28291341  0.13356456  0.913250802
                               family_history_with_overweight         FAVC        FCVC          NCP
Gender                                          -0.10251213 -0.064933774  0.27450478 -0.067599988
Age                                              0.20572533  0.063901686  0.01629089 -0.043943727
Height                                           0.24768389  0.178363778 -0.03812106  0.243671726
Weight                                           0.49682038  0.272300490  0.21612471  0.107468988
family_history_with_overweight                   1.00000000  0.208035507  0.04037225  0.071369697
FAVC                                             0.20803551  1.000000000 -0.02728308 -0.006999943
FCVC                                             0.04037225 -0.027283080  1.00000000  0.042216296
NCP                                              0.07136970 -0.006999943  0.04221630  1.000000000
CAEC                                            -0.16978653 -0.150067628  0.05467024  0.097801337
SMOKE                                            0.01738550 -0.050659965  0.01431953  0.007811192
CH2O                                             0.14743661  0.009719131  0.06846147  0.057087996
SCC                                             -0.18542171 -0.190658309  0.07185219 -0.015623955
FAF                                             -0.05667320 -0.107995159  0.01993940  0.129504307
TUE                                              0.02294330  0.068416912 -0.10113485  0.036325572
CALC                                            -0.03667591  0.089519515  0.06078109  0.071746765
MTRANS                                          -0.09922900 -0.071443854  0.06374632 -0.052962132
NObeyesdad                                       0.50514842  0.247793200  0.22775883  0.026690439
                                      CAEC        SMOKE         CH2O          SCC         FAF
  Gender                          0.09154334 -0.044698091 -0.107929676  0.102633482 -0.18960696
  Age                            -0.08373870  0.091987445 -0.045303858 -0.116282847 -0.14493833
  Height                         -0.04881820  0.055499384  0.213375917 -0.133752777  0.29470900
  Weight                         -0.28749346  0.025746413  0.200575387 -0.201906340 -0.05143627
  family_history_with_overweight -0.16978653  0.017385500  0.147436606 -0.185421707 -0.05667320
  FAVC                           -0.15006763 -0.050659965  0.009719131 -0.190658309 -0.10799516
  FCVC                            0.05467024  0.014319529  0.068461472  0.071852192  0.01993940
  NCP                             0.09780134  0.007811192  0.057087996 -0.015623955  0.12950431
  CAEC                            1.00000000  0.055281967 -0.144995140  0.109178952  0.03011022
  SMOKE                           0.05528197  1.000000000 -0.031994530  0.047731227  0.01121603
  CH2O                           -0.14499514 -0.031994530  1.000000000  0.008036485  0.16723649
  SCC                             0.10917895  0.047731227  0.008036485  1.000000000  0.07422066
  FAF                             0.03011022  0.011216029  0.167236492  0.074220664  1.00000000
  TUE                             0.04856704  0.017613134  0.011965338 -0.010927978  0.05856207
  CALC                           -0.04753960  0.082471289  0.091385557  0.003462876 -0.08679871
  MTRANS                          0.04493057 -0.013098306  0.045227319  0.041639660  0.01037684
  NObeyesdad                     -0.32934972  0.003442179  0.133008436 -0.194507656 -0.19990084
                                        TUE         CALC       MTRANS   NObeyesdad
  Gender                         -0.01726947  0.007615872  0.137378117  0.031463618
  Age                            -0.29693059  0.044487106 -0.602316964  0.282913410
  Height                          0.05191167  0.129731863 -0.071615927  0.133564558
  Weight                         -0.07156136  0.206676696  0.005742124  0.913250802
  family_history_with_overweight  0.02294330 -0.036675907 -0.099229000  0.505148418
  FAVC                            0.06841691  0.089519515 -0.071443854  0.247793200
  FCVC                           -0.10113485  0.060781094  0.063746319  0.227758827
  NCP                             0.03632557  0.071746765 -0.052962132  0.026690439
  CAEC                            0.04856704 -0.047539605  0.044930568 -0.329349716
  SMOKE                           0.01761313  0.082471289 -0.013098306  0.003442179
  CH2O                            0.01196534  0.091385557  0.045227319  0.133008436
  SCC                            -0.01092798  0.003462876  0.041639660 -0.194507656
  FAF                             0.05856207 -0.086798709  0.010376839 -0.199900835
  TUE                             1.00000000 -0.045864068  0.179617545 -0.107991495
  CALC                           -0.04586407  1.000000000  0.013113489  0.151752322
  MTRANS                          0.17961755  0.013113489  1.000000000  0.012269058
  NObeyesdad                     -0.10799149  0.151752322  0.012269058  1.000000000
```

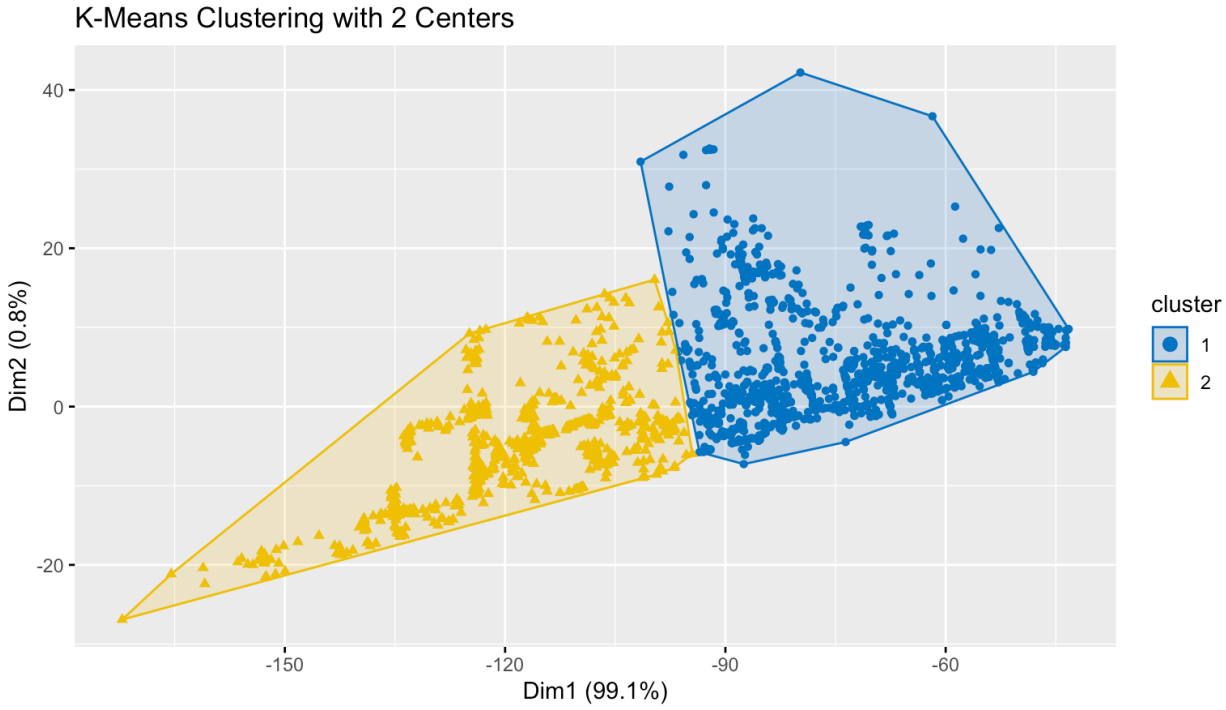


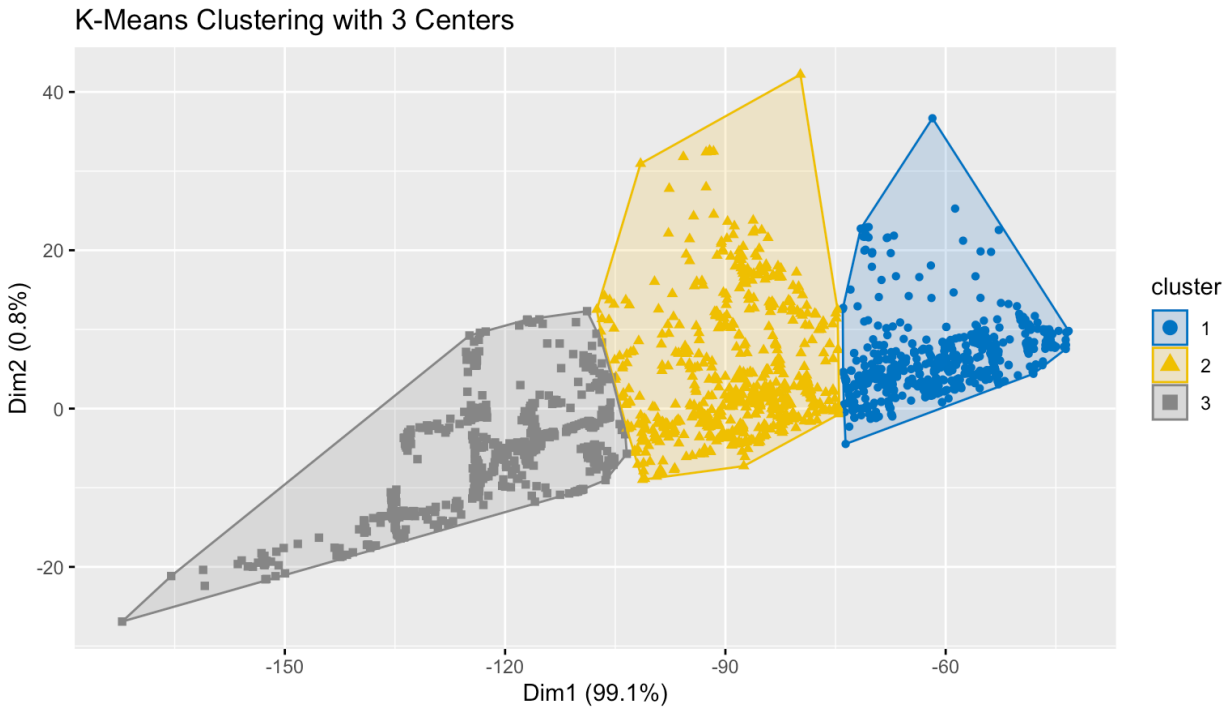
Correlation Plot

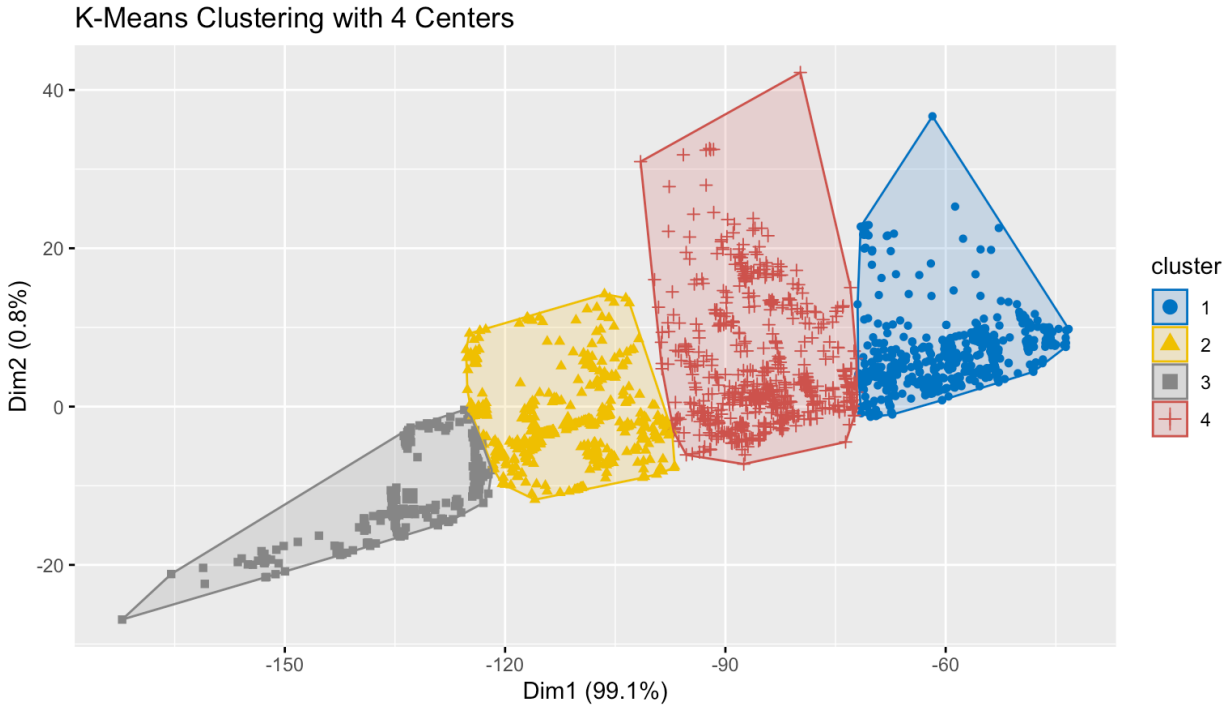
K Means

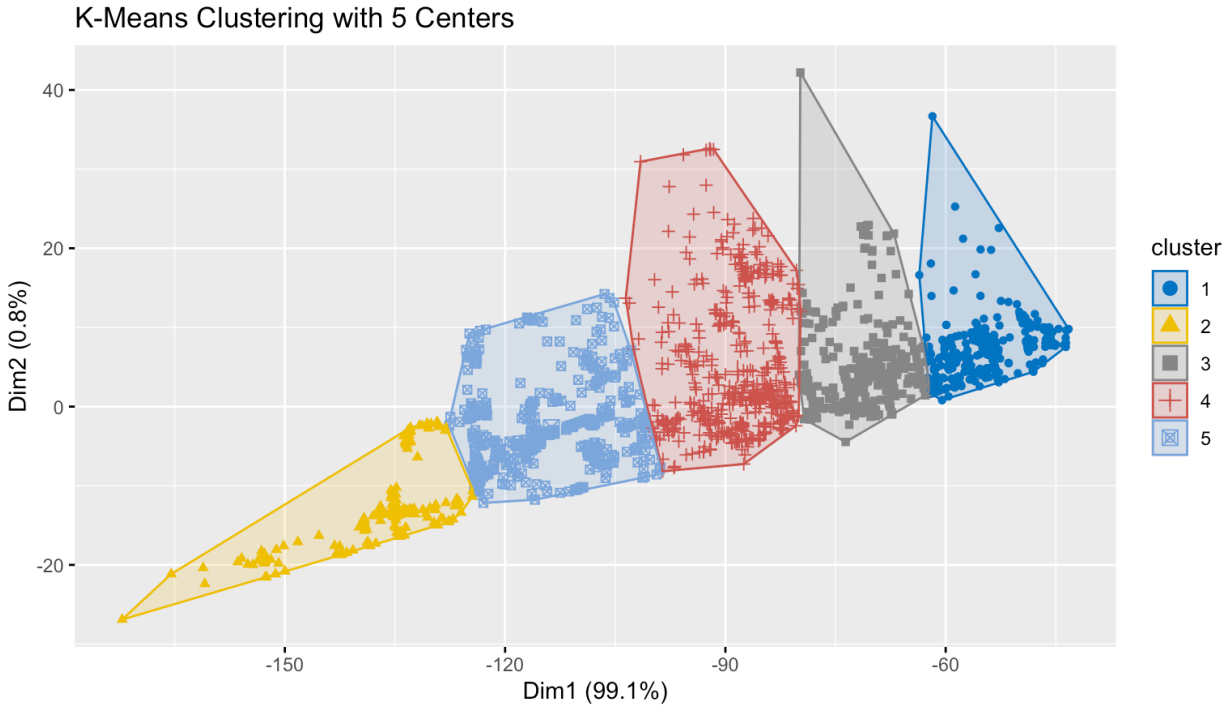
We are going to perform clustering using K Means, aiming to partition observations into a specified number of clusters. The function takes two arguments: data, representing the dataset to be clustered, and centers, indicating the number of cluster centers to create. Inside the function, it first selects the numeric columns from the dataset, as K-means clustering works only with numeric data. Then, it applies the kmeans() function to perform clustering on the numeric data, specifying the number of desired centers and the number of random sets to be used (nstart = 25). The resulting clustering results are printed, and then a visualization of the clusters is created using the fviz_cluster() function from the "factoextra" package. This function plots the clusters as points in a scatterplot, with ellipses representing the cluster boundaries.

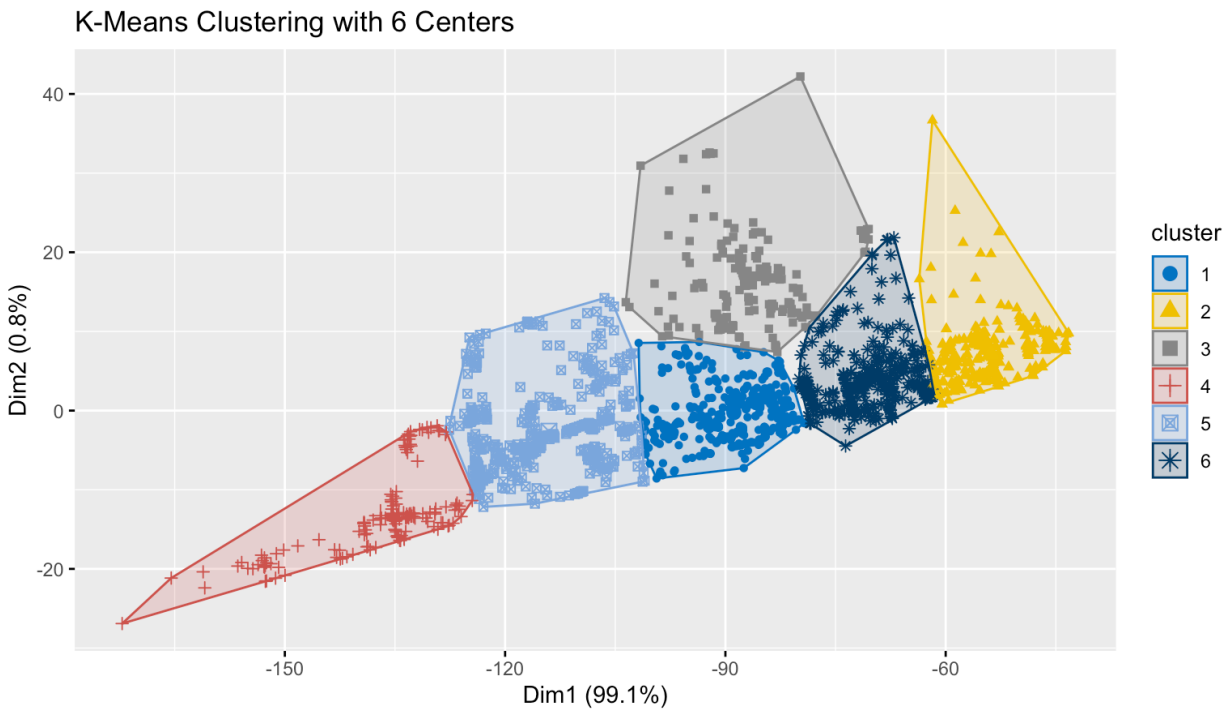
The subsequent lines of code call the perform_clustering function multiple times with different numbers of centers, ranging from 2 to 6. We are Increasing the number of centers to reveal finer patterns or structures within the data, but it also risks overfitting or creating overly complex interpretations.


K-Means Clustering with 2 Centers
Dim2 (0.8%)
40
20
0
-20
-150
-120
-90
-60
Dim1 (99.1%)
cluster
1
2


K-Means Clustering with 3 Centers
Dim2 (0.8%)
40
20
0
-20
-150
-120
-90
-60
Dim1 (99.1%)
cluster
1
2
3


K-Means Clustering with 4 Centers
Dim2 (0.8%)
40
20
0
-20
-150
-120
-90
-60
Dim1 (99.1%)
cluster
1
2
3
4

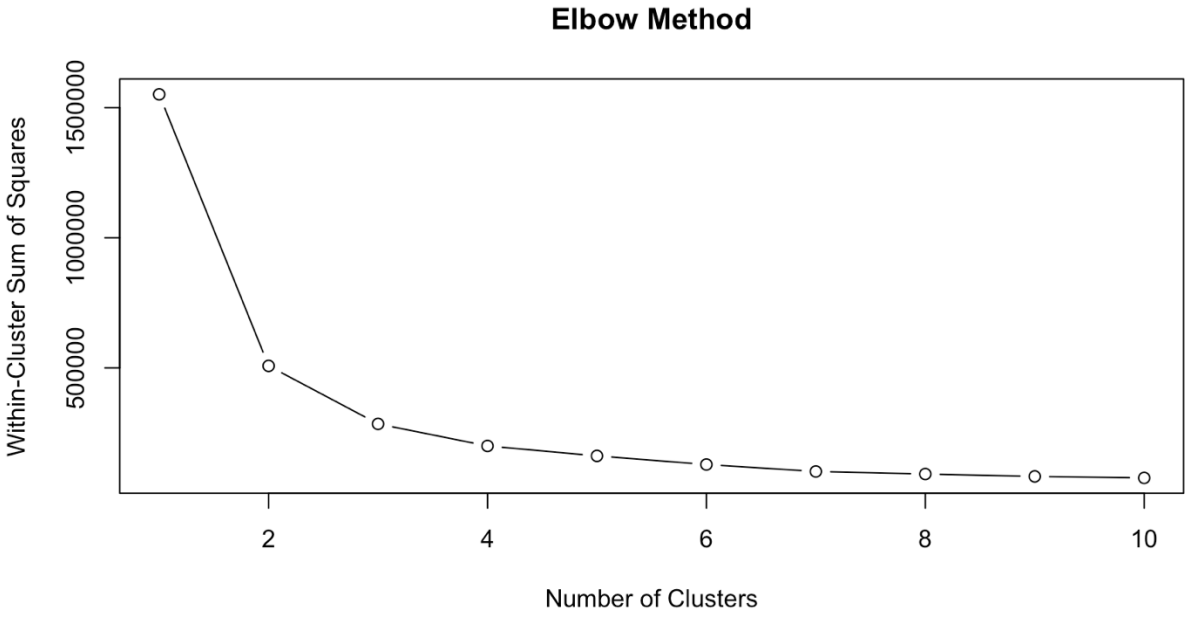

K-Means Clustering with 5 Centers
Dim2 (0.8%)
40
20
0
-20
-150
-120
-90
-60
Dim1 (99.1%)
cluster
1
2
3
4
5

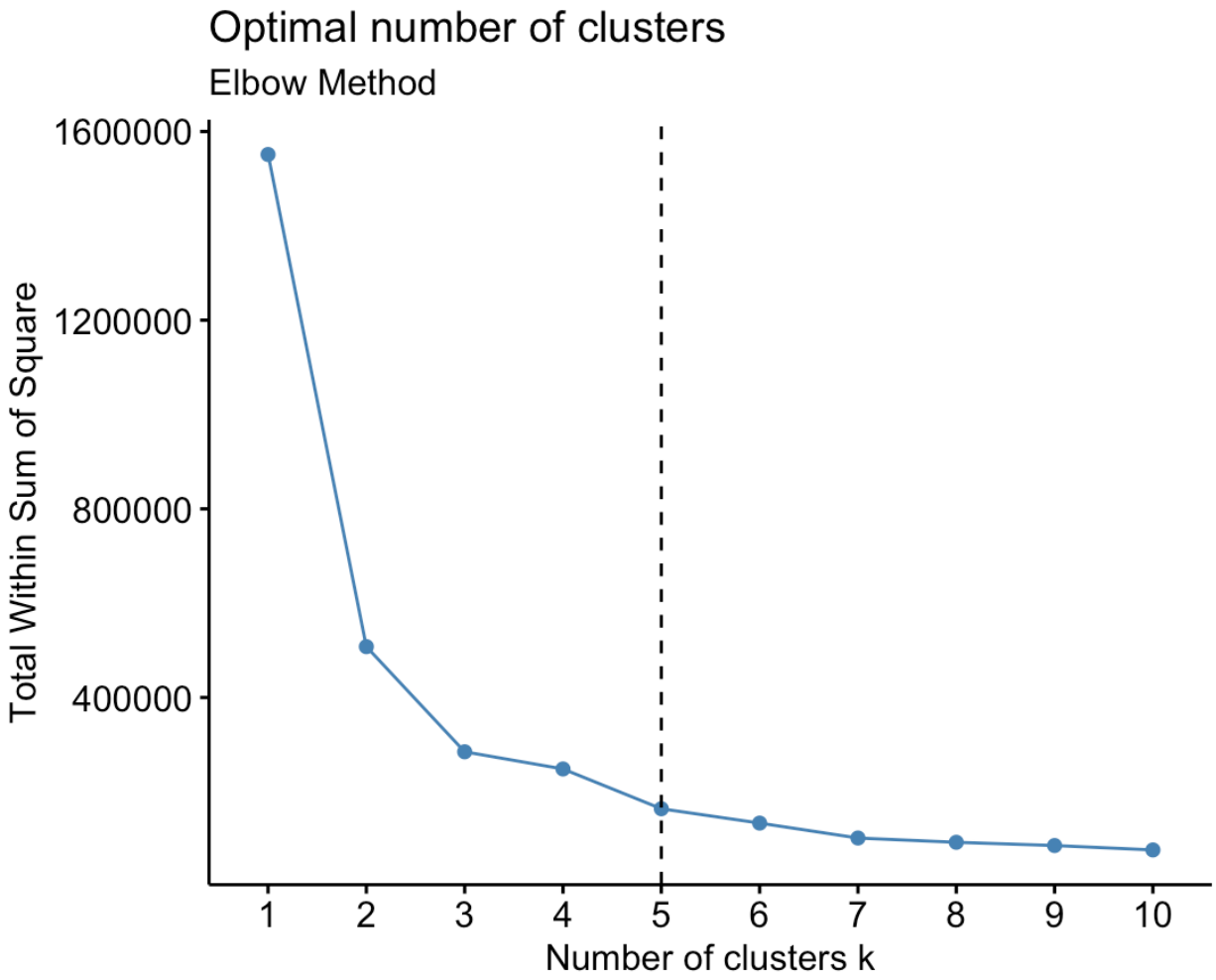


Now we try to determining the optimal number of clusters for K-means clustering using the elbow method. First, we subset the "Obesity" dataset into two subsets based on the presence or absence of family history with overweight. The subset subset_with_history contains observations with a family history of overweight (coded as 2), while subset_without_history contains observations without such a family history (coded as 1).

We then calculate the within-cluster sum of squares (WCSS) for a range of cluster numbers (from 1 to 10) using K-means clustering. This is done by iterating through each value of k (number of clusters) and applying K-means clustering to the numeric columns of the dataset. The WCSS for each value of k is stored in the vector wcss.

After computing WCSS for each k, we generate elbow plot using plot() to visualize the relationship between the number of clusters and the WCSS. Finally, we utilize the fviz_nbclust() function from the "factoextra" package to further assess the optimal number of clusters. This function generates a plot showing various methods for determining the number of clusters, including the within-cluster sum of squares method. It also adds a vertical dashed line at the identified optimal number of clusters assisting in visualizing the elbow point on the plot.


Elbow Method
Within-Cluster Sum of Squares
500000
1000000
1500000
2
4
6
8
10
Number of Clusters


Optimal number of clusters
Elbow Method
Total Within Sum of Square
1600000
1200000
800000
400000
1
2
3
4
5
6
7
8
9
10
Number of clusters k

KNN

Now we use KNN to provide supervised predictions of cluster labels for the testing data, leveraging the clustering information obtained from the training data through K-Means. This serves as a means to evaluate the clustering performance and to compare the clustering results with other methods or against known labels. First, we split the data into training and testing sets using a 70-30 ratio. We use Random sampling to select the training indices. Then KNN classification is performed using the training data features and cluster labels obtained from K-Means clustering. Now we generate a confusion matrix to evaluate the agreement between predicted and refined clusters.

```
> confusionMatrix(as.factor(test_data$PredictedCluster), as.factor(tes
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3   4   5
         1   0   0  33 139   0
         2   0   0   0   0  99
         3  54 117   0  12   0
         4   5  52   0   0  57
         5   0   0  66   0   0

Overall Statistics

               Accuracy : 0
                 95% CI : (0, 0.0058)
    No Information Rate : 0.2666
    P-Value [Acc > NIR] : 1

                  Kappa : -0.2201

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity           0.00000   0.0000   0.0000   0.0000   0.0000
Specificity           0.70087   0.7871   0.6579   0.7640   0.8619
Pos Pred Value        0.00000   0.0000   0.0000   0.0000   0.0000
Neg Pred Value        0.87229   0.6841   0.7805   0.7096   0.7254
Prevalence            0.09306   0.2666   0.1562   0.2382   0.2461
Detection Rate        0.00000   0.0000   0.0000   0.0000   0.0000
Detection Prevalence  0.27129   0.1562   0.2886   0.1798   0.1041
Balanced Accuracy     0.35043   0.3935   0.3290   0.3820   0.4310
```

Linear Modeling

Now we perform linear regression analysis to predict obesity levels based on various predictor variables, evaluates the model's performance, and provides diagnostic plots to assess the model's assumptions and validity. We split the data into training and testing sets using a 70-30 ratio. This splitting is done to train the model on one portion of the data and evaluate its performance on unseen data.

lm_model a linear regression model is fitted to the training data using the glm() function. The model aims to predict the "NObeyesdad" variable based on predictor variables such as "NCP", "Weight", "family_history_with_overweight", "MTRANS", "CALC", "FAVC", "FAF", and "SCC". The selection of these attributes is based on their potential relevance in predicting obesity levels. Factors such as dietary habits, physical activity, family history, and lifestyle choices are considered important indicators of obesity and are therefore selected in the analysis model.

```
> summary(lm_model)

Call:
glm(formula = NObeyesdad ~ NCP + Weight + family_history_with_overweight +
    MTRANS + CALC + FAVC + FAF + SCC, family = gaussian(), data = train)

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    -1.2732335  0.2074870  -6.136 1.08e-09 ***
NCP                            -0.1158428  0.0247478  -4.681 3.12e-06 ***
Weight                          0.0679126  0.0008981  75.620  < 2e-16 ***
family_history_with_overweight  0.2782310  0.0580050   4.797 1.78e-06 ***
MTRANS                          0.0170531  0.0152130   1.121 0.262492
CALC                           -0.1444371  0.0384072  -3.761 0.000176 ***
FAVC                           -0.0750309  0.0608338  -1.233 0.217633
FAF                            -0.3528871  0.0228473 -15.445  < 2e-16 ***
SCC                             0.0038159  0.0920926   0.041 0.966954
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.531466)

    Null deviance: 5914.66  on 1476  degrees of freedom
Residual deviance:  780.19  on 1468  degrees of freedom
AIC: 3268.9

Number of Fisher Scoring iterations: 2
```

Then we use anova() function to compute an analysis of variance table for the fitted model. This table assesses the significance of each predictor variable in explaining the variability in the response variable.

```
> print(anova_lm)
Analysis of Deviance Table

Model: gaussian, link: identity

Response: NObeyesdad

Terms added sequentially (first to last)


                                Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                             1476     5914.7
NCP                              1      3.3      1475     5911.3   0.01233 *
Weight                           1   4979.5      1474      931.8 < 2.2e-16 ***
family_history_with_overweight   1     20.2      1473      911.7 7.301e-10 ***
MTRANS                           1      0.8      1472      910.9   0.23370
CALC                             1      3.5      1471      907.5   0.01075 *
FAVC                             1      0.1      1470      907.3   0.64541
FAF                              1    127.2      1469      780.2 < 2.2e-16 ***
SCC                              1      0.0      1468      780.2   0.96695
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

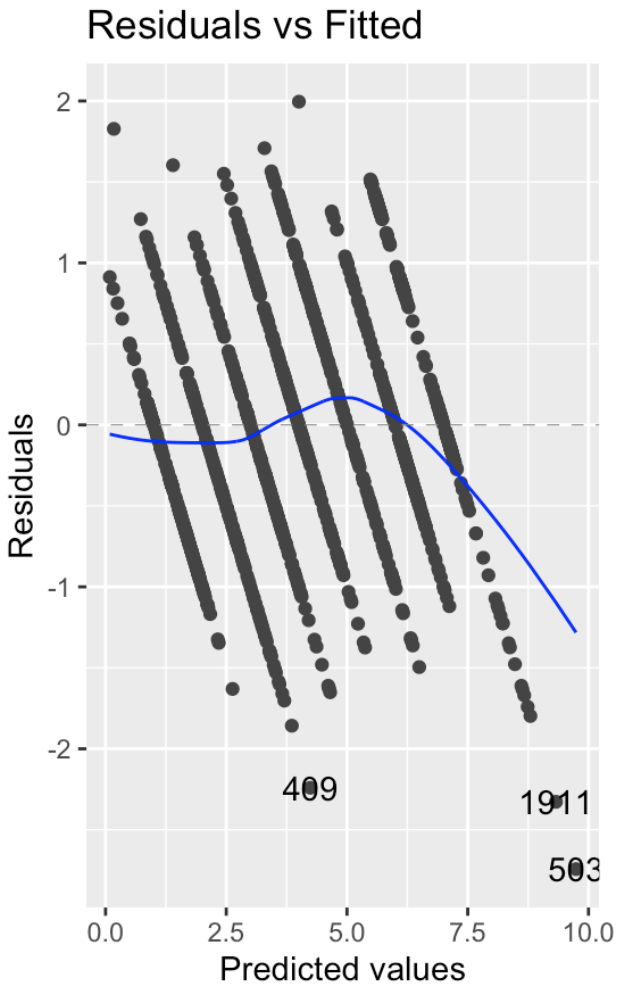
Now Predictions are made on the testing dataset using the fitted linear regression model . The predict() function generates predicted values based on the model and the predictor variables in the testing dataset.

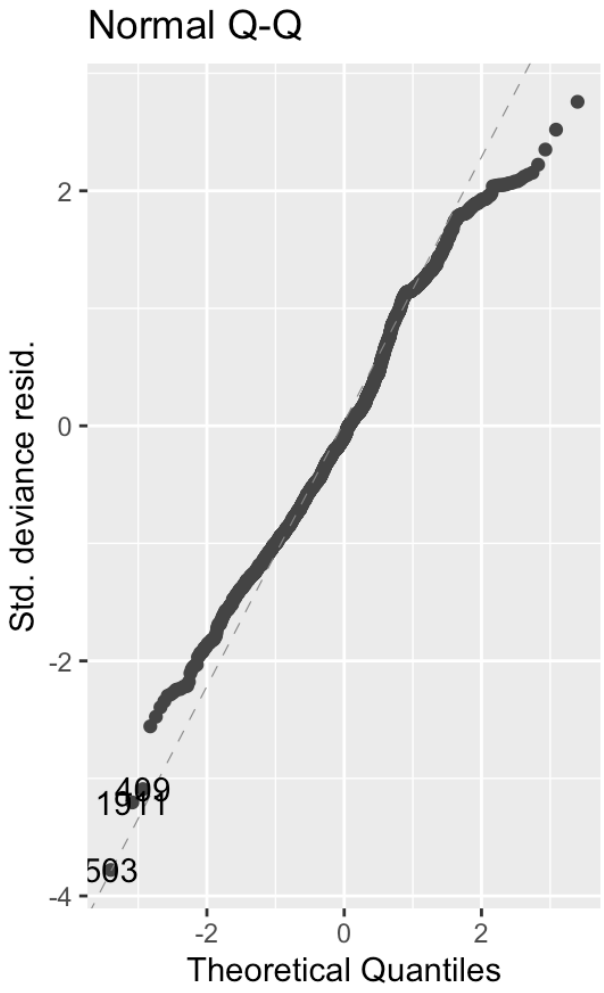
```
> predictions <- predict(lm_model, newdata = test, type = "response")
> # Summary of predictions
> summary(predictions)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1503  2.8397  4.0173  4.1701  5.7374  9.6119
```

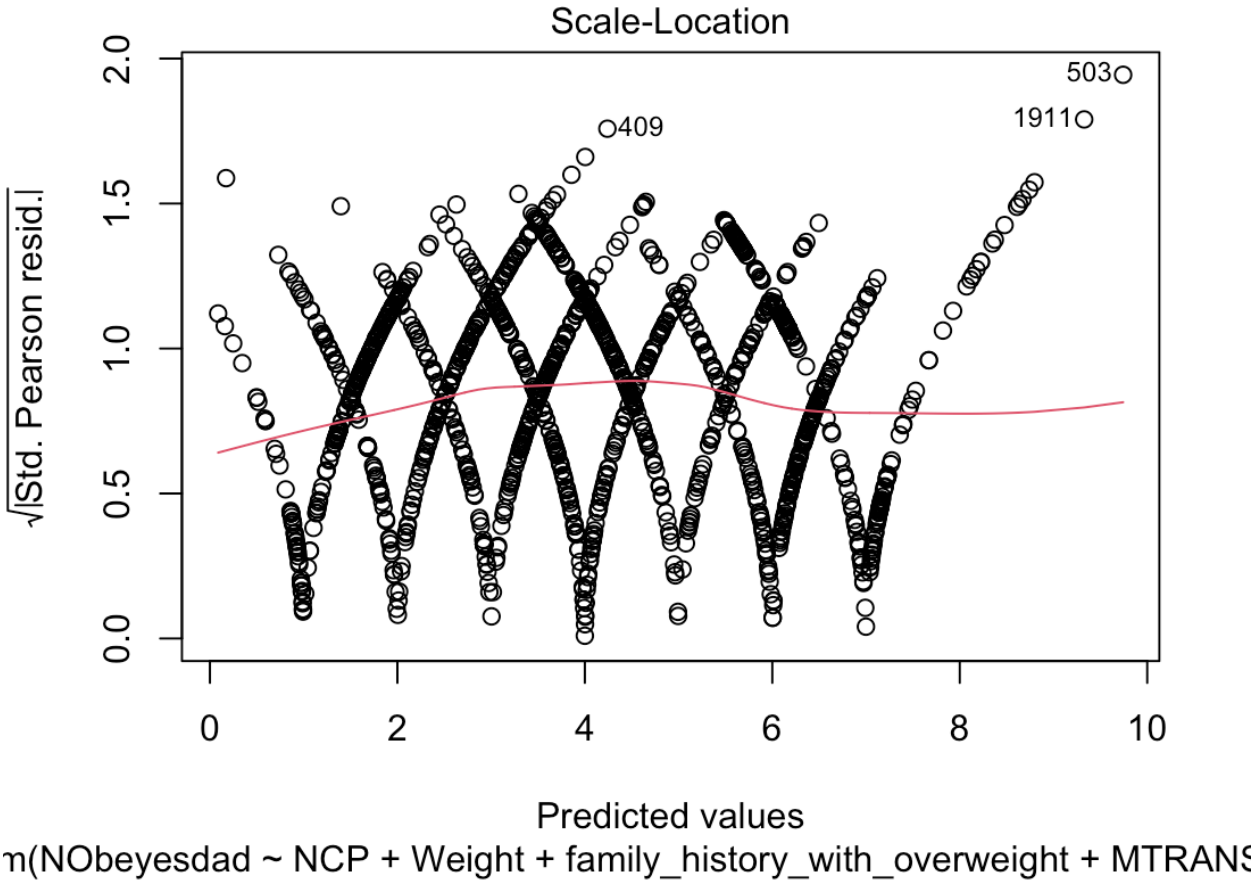
Confidence intervals for the coefficients of the linear regression model are computed using the confint() function. These intervals provide a range of plausible values for the coefficients with a specified level of confidence.

```
> head(test)
   Gender Age Height Weight family_history_with_overweight FAVC FCVC NCP CAEC SMOKE CH2O SCC FAF TUE CALC MTRANS
3       1  23   1.80   77.0                               2    1    2   3    2     1    2   1   2   1    3      4
5       1  22   1.78   89.8                               1    1    2   1    2     1    2   1   0   0    2      4
7       2  23   1.50   55.0                               2    2    3   3    2     1    2   1   1   0    2      2
15      1  23   1.77   60.0                               2    2    3   1    2     1    1   1   1   1    2      4
21      1  22   1.65   80.0                               2    1    2   3    2     1    2   1   3   2    1      5
22      2  52   1.69   87.0                               2    2    3   1    2     2    2   1   0   0    1      1
   NObeyesdad predicted_NObeyesdad
3           2             3.022884
5           4             4.695831
7           2             1.916993
15          2             2.522348
21          4             3.179661
22          5             4.802154
```

Lastly we generate several diagnostic plots to assess the assumptions and performance of the linear regression model. These plots include Residuals vs. Fitted plot, Normal Q-Q plot, and Scale-Location plot.




Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
2
0
-2
-4
-2
0
2
409
1911
503



Discussion

The analysis's conclusions include information about variables including the amount of time spent using technology, drinking water, eating habits, frequency of physical activity, and others that are linked to obesity levels. Different groups within the dataset are shown by clustering analysis, which may indicate various obesity risk profiles. With clusters serving as characteristics, the KNN classification model offers a potential method for somewhat accurately predicting obesity levels. Furthermore, by shedding light on the correlation between predictor variables and obesity levels, linear modeling facilitates a deeper comprehension of the variables that contribute to obesity. All things considered, the research provides insightful information about the obesity dataset, helping to uncover hidden patterns and influencing future interventions and obesity prevention strategies.