CLASS PROJECT 2

EXPLORING VARIATIONS IN CLUSTERING AND PREDICTIVE ANALYSIS

CSCI 6444: INTRODUCTION TO BIG DATA & ANALYTICS

PROFESSOR: STEPHEN KAISLER

GROUP 12

NAME: RAJITH RAVIKUMAR

GWID: G45692898

NAME: SRAVANI BRAHAMMA ROUTHU

GWID: G28059824

DATASET DESCRIPTION

The dataset contains a mix of raw and synthetic data related to obesity and health factors. It has 2112 instances (observations) and 17 attributes (features). The attributes include a mix of categorical and numeric variables related to an individual's demographic information, lifestyle, eating habits, physical activity, and body mass index (BMI) category.

Attribute information:

Variable Name	Role	Type	Demographic	Description	Units	Missing Values
Gender	Feature	Categorical	Gender			no
Age	Feature	Continuous	Age			no
Height	Feature	Continuous				no
Weight	Feature	Continuous				no
family_history_with_overweight	Feature	Binary		Has a family member suffered or suffers from overweight?		no
FAVC	Feature	Binary		Do you eat high caloric food frequently?		no
FCVC	Feature	Integer		Do you usually eat vegetables in your meals?		no
NCP	Feature	Continuous		How many main meals do you have daily?		no
CAEC	Feature	Categorical		Do you eat any food between meals?		no
SMOKE	Feature	Binary		Do you smoke?		no
CH2O	Feature	Continuous		How much water do you drink daily?		no
SCC	Feature	Binary		Do you monitor the calories you eat daily?		no
FAF	Feature	Continuous		How often do you have physical activity?		no
TUE	Feature	Integer		How much time do you use technological devices such as cell phone, videogames, television, computer and others?		no
CALC	Feature	Categorical		How often do you drink alcohol?		no
MTRANS	Feature	Categorical		Which transportation do you usually use?		no
NObeyesdad	Target	Categorical		Obesity level		no

OBJECTIVE

The project aims to predict which features are the most useful in predicting obesity

1. PREPARING THE DATA AND PERFORMING PAIRWISE PLOTTING

We start by importing our dataset using the **read.table** method and using the comma sign "," as a separator, into the **obesity** variable. We can see that the feature names are from V1 to V17.

```
> obesity_data_file<- file.path(directory,"ObesityDataSet_raw_and_data_sinthetic.csv")
> obesity<-read.table(file = obesity_data_file, stringsAsFactors = FALSE,sep = ',')
> obesity[1:10,]
       V1  V2     V3     V4                           V5   V6   V7  V8        V9   V10  V11 V12 V13 V14        V15                  V16
1  Gender Age Height Weight family_history_with_overweight FAVC FCVC NCP      CAEC SMOKE CH2O SCC FAF TUE       CALC               MTRANS
2  Female  21   1.62     64                          yes   no    2   3 Sometimes    no    2  no   0   1         no Public_Transportation
3  Female  21   1.52     56                          yes   no    3   3 Sometimes   yes    3 yes   3   0  Sometimes Public_Transportation
4    Male  23    1.8     77                          yes   no    2   3 Sometimes    no    2  no   2   1 Frequently Public_Transportation
5    Male  27    1.8     87                           no   no    3   3 Sometimes    no    2  no   2   0 Frequently               Walking
6    Male  22   1.78   89.8                           no   no    2   1 Sometimes    no    2  no   0   0  Sometimes Public_Transportation
7    Male  29   1.62     53                           no  yes    2   3 Sometimes    no    2  no   0   0  Sometimes            Automobile
8  Female  23    1.5     55                          yes  yes    3   3 Sometimes    no    2  no   1   0  Sometimes             Motorbike
9    Male  22   1.64     53                           no   no    2   3 Sometimes    no    2  no   3   0  Sometimes Public_Transportation
10   Male  24   1.78     64                          yes  yes    3   3 Sometimes    no    2  no   1   1 Frequently Public_Transportation
                  V17
1           NObeyesdad
2        Normal_Weight
3        Normal_Weight
4        Normal_Weight
5   Overweight_Level_I
6  Overweight_Level_II
7        Normal_Weight
8        Normal_Weight
9        Normal_Weight
10       Normal_Weight
>
```

To fix this we can use the "head = TRUE" to use the first row of the csv file as the feature names as shown below.

```
> library(igraph)
> directory<-"C:/Users/rajit/Downloads"
> obesity_data_file<- file.path(directory,"ObesityDataSet_raw_and_data_sinthetic.csv")
> obesity<-read.table(file = obesity_data_file, stringsAsFactors = FALSE,sep = ',', header = TRUE)
```

```
> dim(obesity)
[1] 2111   17
> obesity[1:10,]
   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
>
```

We can proceed to import the following libraries which will prove useful for our analysis: "**sna**", "**psych**", "**factoextra**", "**plotly**", "**ggplot2**", "**corrplot**", **"class"** and "**lattice**".

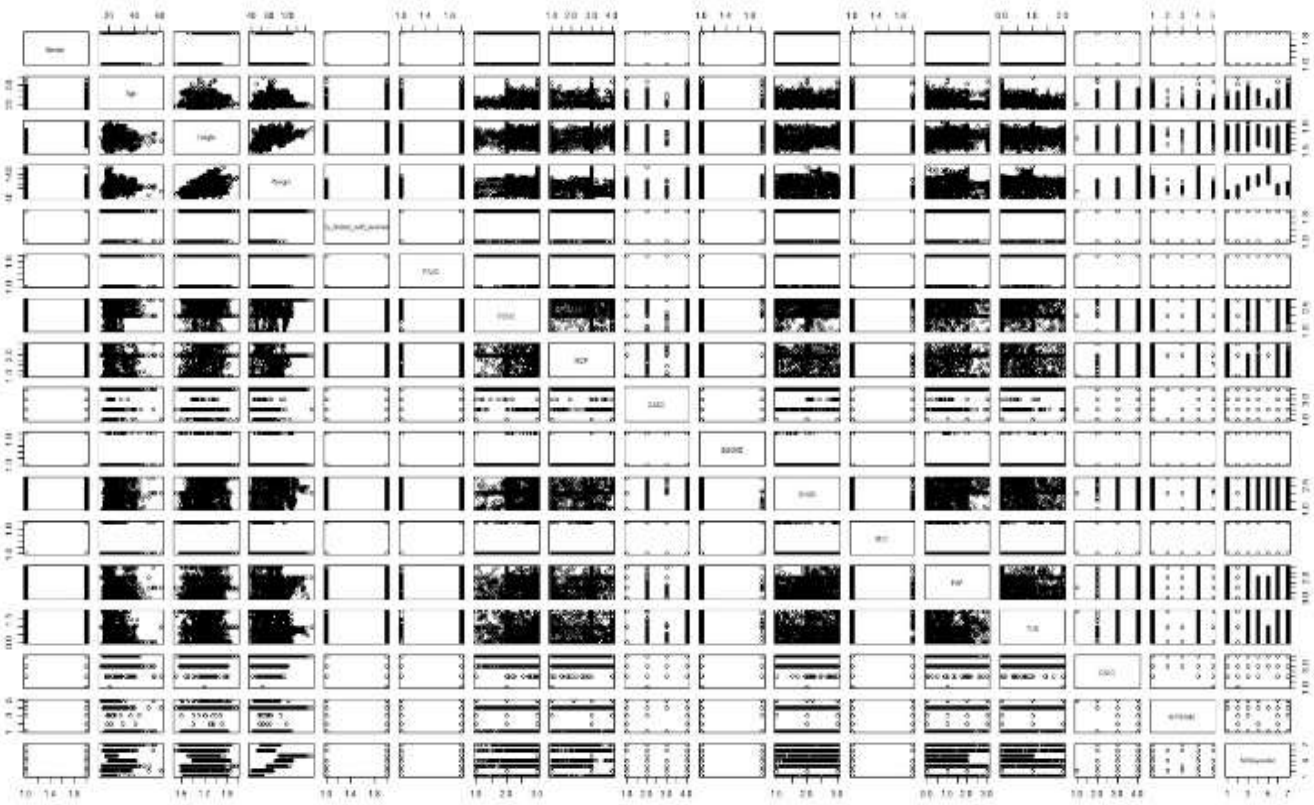
```
library(igraph)
library(sna)
library(psych)
library(factoextra)
library(plotly)
library(ggplot2)
library(corrplot)
library(lattice)
```

1a) For this data set, plot the data using pairwise plotting to get a sense of the relationships between the attributes.

Try plotting the data using several plotting functions to see what it looks like. Use pairs (e.g., 2D plots) or 3 variables (3D plots) based on the packages.

- We first proceed to plot all the 17 variables in a pairwise manner.

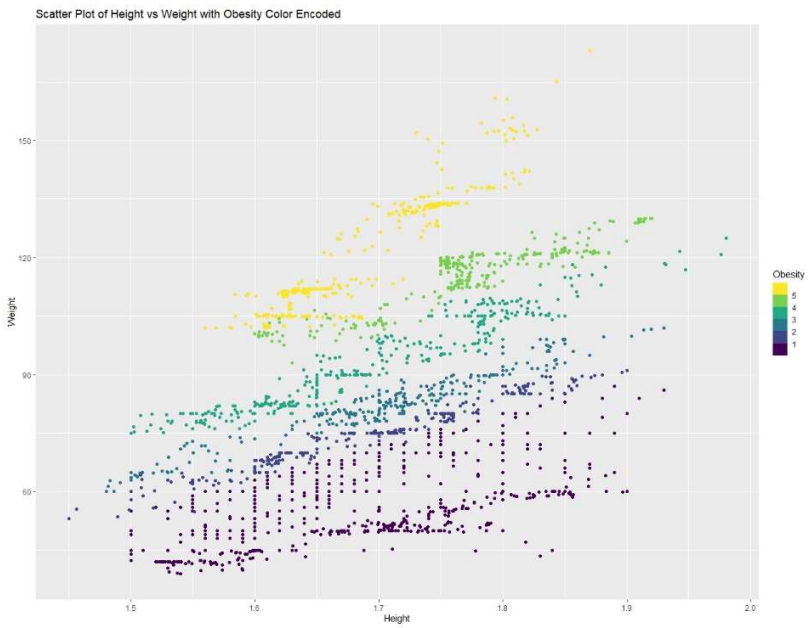
```
> plot(obesity)
>
```

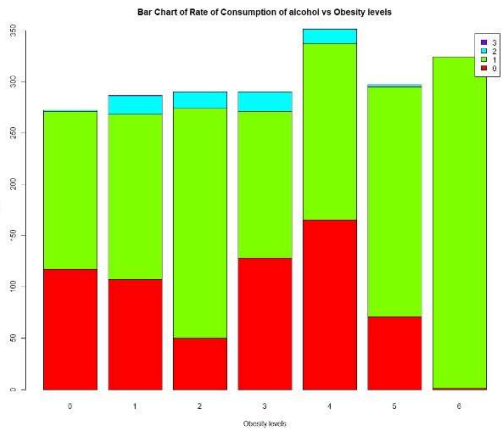


1b) Which pairs of attributes seem to be correlated? How are they correlated?

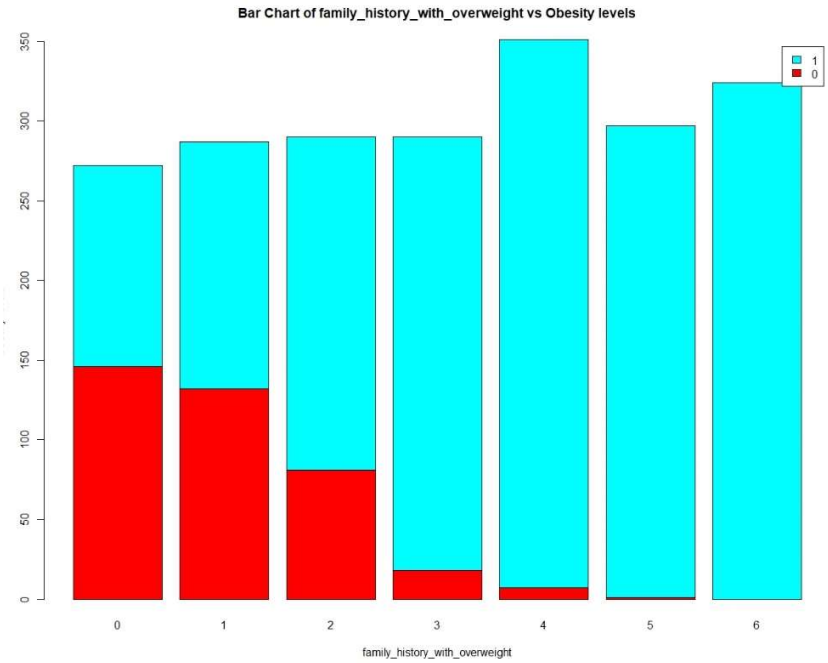
As shown in the scatter plot below height and weight seems to be linearly correlated.

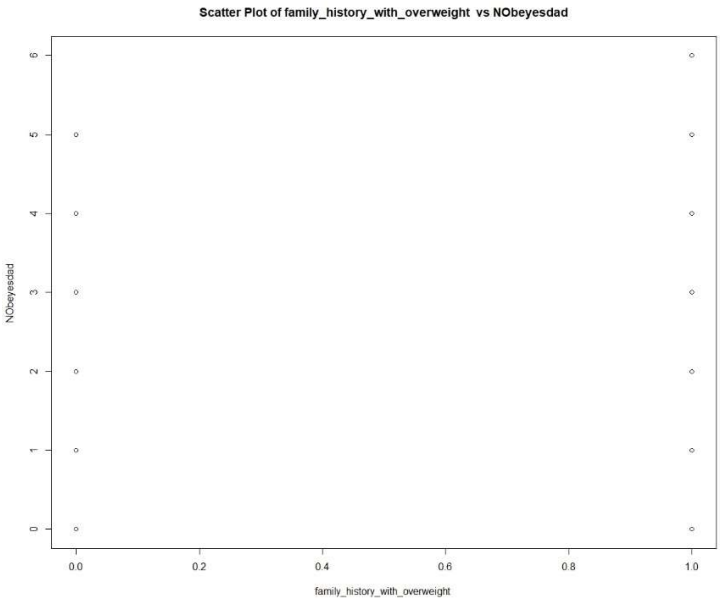
```
#Scatter Plot between height and weight
library(ggplot2)

ggplot(obesity_encoded, aes(x = Height, y = Weight, color = NObeyesdad)) +
  geom_point() +
  scale_color_viridis_b()+
  labs(x = "Height", y = "Weight", color = "Obesity") +
  ggtitle("Scatter Plot of Height vs Weight with Obesity Color Encoded")

#
```



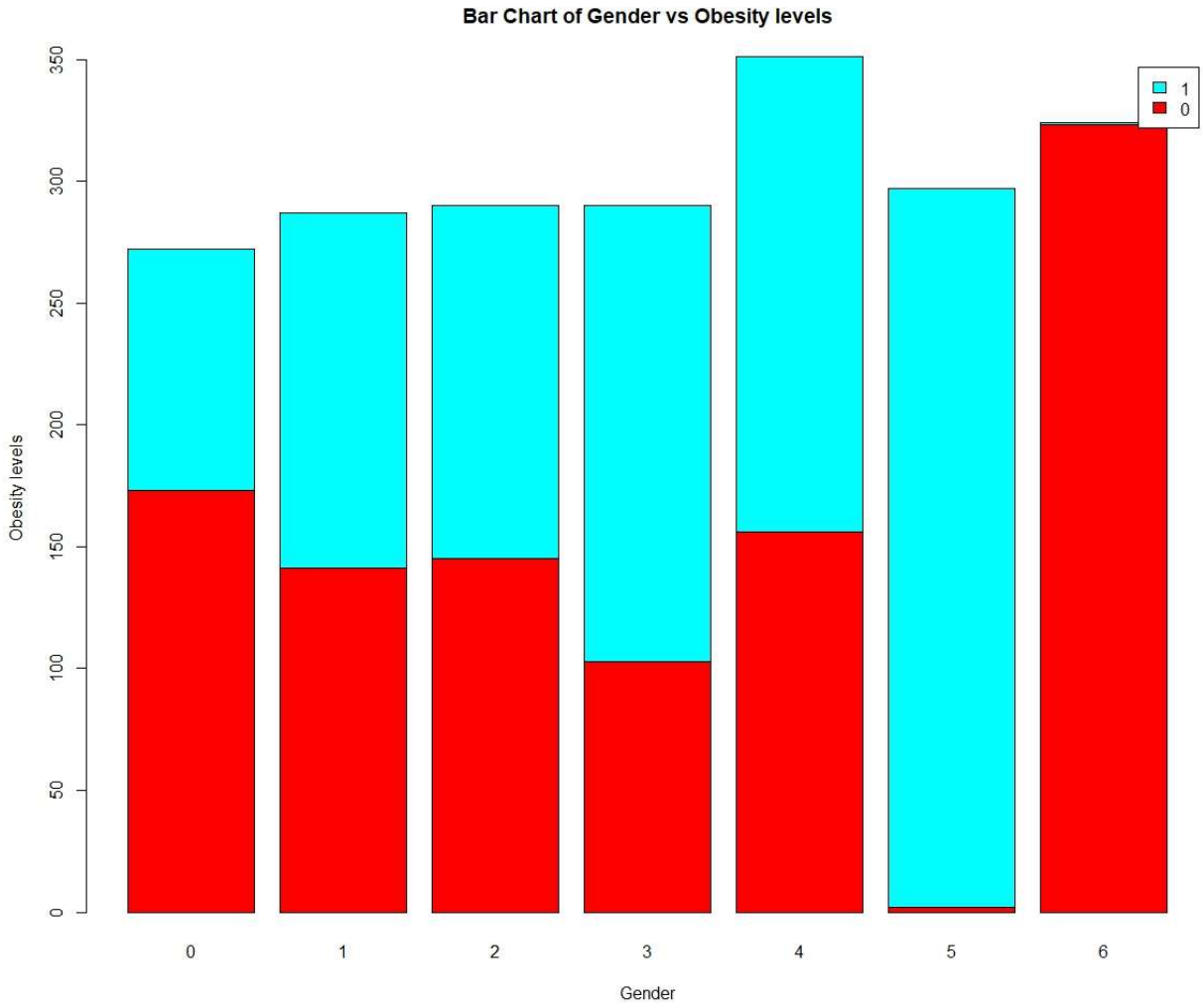


From the above scatter plot we can see that people who do not consume alcohol are close to 0 in Obesity_Type_III and all "Overweight_Level_I", "Overweight_Level_II", "Obesity_Type_I", "Obesity_Type_II", "Obesity_Type_III" categories are dominated by people who consume alcohol.





From the images above we can see that people with no familial history of being overweight have never been of Obesity_Type_III. The distribution of people in the insufficient weight and normal weight are evenly distributed among people with no familial history of overweight and people with familial history of overweight. On the other hand, "Overweight_Level_I", "Overweight_Level_II", "Obesity_Type_I", "Obesity_Type_II", "Obesity_Type_III" categories are dominated by people with familial history of overweight. Which proves that most causes of high levels of obesity are caused genetically.



The above graph shows that "Obesity_Type_II" is dominated by males and "Obesity_Type_III" is dominated by females. Also insufficient weight category is mostly women.

2. PREPARE THE DATA

2a) Investigate some of the statistics of the data set: summary(). Describe(). What do you glean from this data?
We then apply the describe() and summary() methods to get a better sense of our data.

```
> describe(obesity)
                                vars    n  mean    sd median trimmed   mad   min    max  range  skew kurtosis   se
Gender*                            1 2111  1.51  0.50   2.00    1.51  0.00  1.00   2.00   1.00 -0.02    -2.00 0.01
Age                                2 2111 24.31  6.35  22.78   23.34  4.78 14.00  61.00  47.00  1.53     2.81 0.14
Height                             3 2111  1.70  0.09   1.70    1.70  0.10  1.45   1.98   0.53 -0.01    -0.57 0.00
Weight                             4 2111 86.59 26.19  83.00   85.82 32.22 39.00 173.00 134.00  0.26    -0.70 0.57
family_history_with_overweight*    5 2111  1.82  0.39   2.00    1.90  0.00  1.00   2.00   1.00 -1.64     0.70 0.01
FAVC*                              6 2111  1.88  0.32   2.00    1.98  0.00  1.00   2.00   1.00 -2.40     3.74 0.01
FCVC                               7 2111  2.42  0.53   2.39    2.46  0.57  1.00   3.00   2.00 -0.43    -0.64 0.01
NCP                                8 2111  2.69  0.78   3.00    2.77  0.00  1.00   4.00   3.00 -1.11     0.38 0.02
CAEC*                              9 2111  3.67  0.78   4.00    3.87  0.00  1.00   4.00   3.00 -2.13     3.06 0.02
SMOKE*                            10 2111  1.02  0.14   1.00    1.00  0.00  1.00   2.00   1.00  6.70    42.95 0.00
CH2O                              11 2111  2.01  0.61   2.00    2.01  0.67  1.00   3.00   2.00 -0.10    -0.88 0.01
SCC*                              12 2111  1.05  0.21   1.00    1.00  0.00  1.00   2.00   1.00  4.36    17.02 0.00
FAF                               13 2111  1.01  0.85   1.00    0.94  1.19  0.00   3.00   3.00  0.50    -0.62 0.02
TUE                               14 2111  0.66  0.61   0.63    0.59  0.72  0.00   2.00   2.00  0.62    -0.55 0.01
CALC*                             15 2111  3.63  0.55   4.00    3.70  0.00  1.00   4.00   3.00 -1.17     0.46 0.01
MTRANS*                           16 2111  3.37  1.26   4.00    3.55  0.00  1.00   5.00   4.00 -1.28    -0.20 0.03
NObeyesdad*                       17 2111  4.02  1.95   4.00    4.02  2.97  1.00   7.00   6.00  0.01    -1.19 0.04
> summary(obesity)
    Gender             Age            Height           Weight       family_history_with_overweight      FAVC               FCVC            NCP            CAEC              SMOKE              CH2O
 Length:2111        Min.   :14.00   Min.   :1.450   Min.   : 39.00   Length:2111                    Length:2111        Min.   :1.000   Min.   :1.000   Length:2111        Length:2111        Min.   :1.000
 Class :character   1st Qu.:19.95   1st Qu.:1.630   1st Qu.: 65.47   Class :character               Class :character   1st Qu.:2.000   1st Qu.:2.659   Class :character   Class :character   1st Qu.:1.585
 Mode  :character   Median :22.78   Median :1.700   Median : 83.00   Mode  :character               Mode  :character   Median :2.386   Median :3.000   Mode  :character   Mode  :character   Median :2.000
                    Mean   :24.31   Mean   :1.702   Mean   : 86.59                                                      Mean   :2.419   Mean   :2.686                                         Mean   :2.008
                    3rd Qu.:26.00   3rd Qu.:1.768   3rd Qu.:107.43                                                      3rd Qu.:3.000   3rd Qu.:3.000                                         3rd Qu.:2.477
                    Max.   :61.00   Max.   :1.980   Max.   :173.00                                                      Max.   :3.000   Max.   :4.000                                         Max.   :3.000
     SCC                FAF              TUE              CALC              MTRANS           NObeyesdad
 Length:2111        Min.   :0.0000   Min.   :0.0000   Length:2111        Length:2111        Length:2111
 Class :character   1st Qu.:0.1245   1st Qu.:0.0000   Class :character   Class :character   Class :character
 Mode  :character   Median :1.0000   Median :0.6253   Mode  :character   Mode  :character   Mode  :character
                    Mean   :1.0103   Mean   :0.6579
                    3rd Qu.:1.6667   3rd Qu.:1.0000
                    Max.   :3.0000   Max.   :2.0000
>
```

```
> describe(obesity_encoded)
                                vars    n  mean    sd median trimmed   mad   min    max  range  skew kurtosis   se
Gender                             1 2111  0.51  0.50   1.00    0.51  0.00  0.00   1.00   1.00 -0.02    -2.00 0.01
Age                                2 2111 24.31  6.35  22.78   23.34  4.78 14.00  61.00  47.00  1.53     2.81 0.14
Height                             3 2111  1.70  0.09   1.70    1.70  0.10  1.45   1.98   0.53 -0.01    -0.57 0.00
Weight                             4 2111 86.59 26.19  83.00   85.82 32.22 39.00 173.00 134.00  0.26    -0.70 0.57
family_history_with_overweight     5 2111  0.82  0.39   1.00    0.90  0.00  0.00   1.00   1.00 -1.64     0.70 0.01
FAVC                               6 2111  0.88  0.32   1.00    0.98  0.00  0.00   1.00   1.00 -2.40     3.74 0.01
FCVC                               7 2111  2.42  0.53   2.39    2.46  0.57  1.00   3.00   2.00 -0.43    -0.64 0.01
NCP                                8 2111  2.69  0.78   3.00    2.77  0.00  1.00   4.00   3.00 -1.11     0.38 0.02
CAEC                               9 2111  1.14  0.47   1.00    1.05  0.00  0.00   3.00   3.00  1.90     5.38 0.01
SMOKE                             10 2111  0.02  0.14   0.00    0.00  0.00  0.00   1.00   1.00  6.70    42.95 0.00
CH2O                              11 2111  2.01  0.61   2.00    2.01  0.67  1.00   3.00   2.00 -0.10    -0.88 0.01
SCC                               12 2111  0.05  0.21   0.00    0.00  0.00  0.00   1.00   1.00  4.36    17.02 0.00
FAF                               13 2111  1.01  0.85   1.00    0.94  1.19  0.00   3.00   3.00  0.50    -0.62 0.02
TUE                               14 2111  0.66  0.61   0.63    0.59  0.72  0.00   2.00   2.00  0.62    -0.55 0.01
CALC                              15 2111  0.73  0.52   1.00    0.75  0.00  0.00   3.00   3.00 -0.24    -0.33 0.01
MTRANS                            16 2111  3.66  0.76   4.00    3.81  0.00  0.00   4.00   4.00 -3.25    12.16 0.02
NObeyesdad                        17 2111  3.11  1.99   3.00    3.14  2.97  0.00   6.00   6.00 -0.08    -1.22 0.04
> summary(obesity_encoded)
     Gender           Age            Height           Weight       family_history_with_overweight      FAVC            FCVC            NCP             CAEC            SMOKE              CH2O
 Min.   :0.0000   Min.   :14.00   Min.   :1.450   Min.   : 39.00   Min.   :0.0000                 Min.   :0.0000   Min.   :1.000   Min.   :1.000   Min.   :0.000   Min.   :0.00000   Min.   :1.000
 1st Qu.:0.0000   1st Qu.:19.95   1st Qu.:1.630   1st Qu.: 65.47   1st Qu.:1.0000                 1st Qu.:1.0000   1st Qu.:2.000   1st Qu.:2.659   1st Qu.:1.000   1st Qu.:0.00000   1st Qu.:1.585
 Median :1.0000   Median :22.78   Median :1.700   Median : 83.00   Median :1.0000                 Median :1.0000   Median :2.386   Median :3.000   Median :1.000   Median :0.00000   Median :2.000
 Mean   :0.5059   Mean   :24.31   Mean   :1.702   Mean   : 86.59   Mean   :0.8176                 Mean   :0.8839   Mean   :2.419   Mean   :2.686   Mean   :1.141   Mean   :0.02084   Mean   :2.008
 3rd Qu.:1.0000   3rd Qu.:26.00   3rd Qu.:1.768   3rd Qu.:107.43   3rd Qu.:1.0000                 3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:2.477
 Max.   :1.0000   Max.   :61.00   Max.   :1.980   Max.   :173.00   Max.   :1.0000                 Max.   :1.0000   Max.   :3.000   Max.   :4.000   Max.   :3.000   Max.   :1.00000   Max.   :3.000
      SCC              FAF              TUE              CALC             MTRANS        NObeyesdad
 Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.000
 1st Qu.:0.00000   1st Qu.:0.1245   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:1.000
 Median :0.00000   Median :1.0000   Median :0.6253   Median :1.0000   Median :4.000   Median :3.000
 Mean   :0.04548   Mean   :1.0103   Mean   :0.6579   Mean   :0.7314   Mean   :3.657   Mean   :3.112
 3rd Qu.:0.00000   3rd Qu.:1.6667   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:5.000
 Max.   :1.00000   Max.   :3.0000   Max.   :2.0000   Max.   :3.0000   Max.   :4.000   Max.   :6.000
```

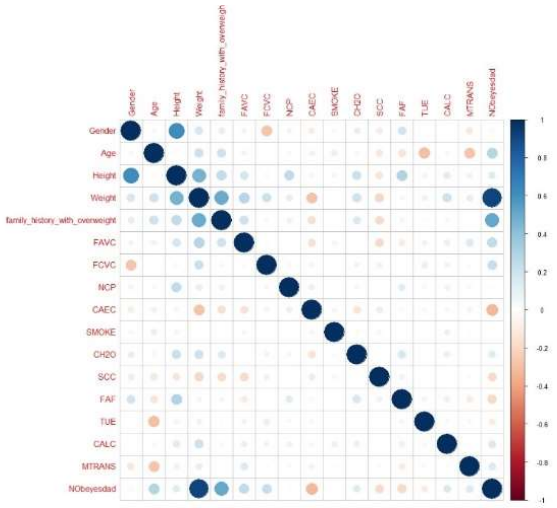
As we can see, there are no NAs in our data frame, which means we do not have to perform extra computation to account for rows with NAs in them, such as deleting those rows or replacing the NA values in a way that accounts for our analysis.

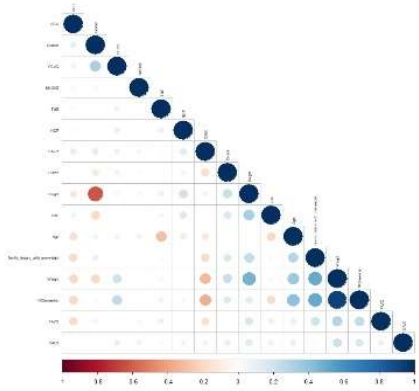
We can observe the mean weight is 86.59Kg and the mean age is 24.31. The Age variable has a skewness of 1.53, indicating a positively skewed distribution with a longer right tail. This could be due to the presence of some older individuals in the dataset. The Weight variable has a maximum value of 173 kg, which is an extremely high weight and could potentially be an outlier or data entry error. The Age variable has a minimum value of 14, which seems unusually low for an obesity study, as it may include children or adolescents.

2b) To subset, pick the most correlated attributes to use – they may all be relevant, so document your rationale for eliminating some attributes. Use at least 5 attributes, but possibly more.

As the rubric points out, correlation matrices are a great way to understand positive and negative correlations between attributes. With the help of the corrplot library, we generate the correlation matrix and plot it as follows:

```
#Correlation without Normalization
obesity_encoded_corr <- cor(obesity_encoded)
obesity_encoded_corr
corrplot::corrplot(obesity_encoded_corr)


#Correlation with Normalization
obesity_encoded_norm <- scale(obesity_encoded[,-16])
pca_res <- prcomp(obesity_encoded_norm)
cor_w <- cor(obesity_encoded_norm, method="spearman")
corrplot(cor_w, type = "lower", order = "hclust", tl.col = "black", tl.cex = 0.5)
```



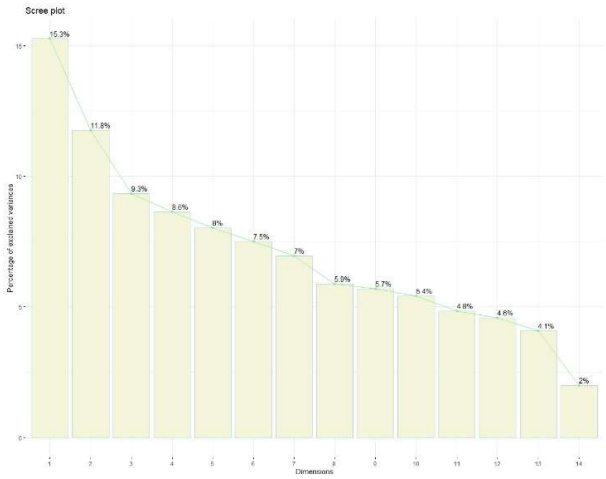


Remove weight and NObeyesdad as they are dependent variables.

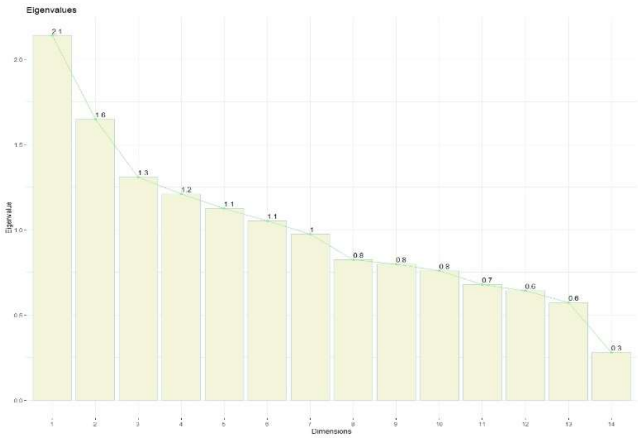
```
#remove weight , NObeyesdad as they are dependent
obesity_encoded_norm <- scale(ObesityDataSet_num[,c(-4,-16,-17)])
pca_res <- prcomp(obesity_encoded_norm)

#the percentage of variances explained by each principal component
fviz_eig(pca_res, addlabels = TRUE, ncp = 15, barfill = "beige", barcolor = "lightblue", linecolor = "lightgreen")

#Eigen Plot
fviz_eig(pca_res, choice = "eigenvalue", ncp = 15, barfill = "beige", barcolor = "lightblue", linecolor = "lightgreen",  addlabels = TRUE,   main = "Eigenvalues")


#Contribution Plot
fviz_contrib(pca_res, "var", axes = 1:6, fill = "beige", color = "lightgreen")
elim_subset_data <- subset(obesity_encoded, select = c("Height","Gender","CH2O", "CALC", "FCVC", "Age", "TUE"))
elim_subset_data
```

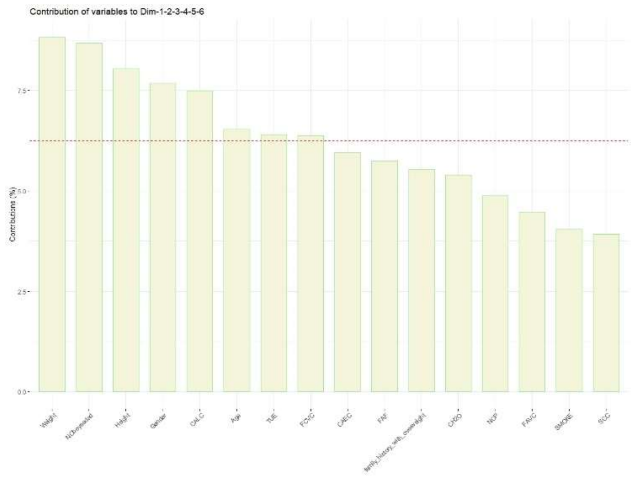
According to the graph below 67.5% of variance can be explained by 7 attributes

1. Gender,
2. Height,
3. Age,
4. FCVC
5. CH20
6. CALC
7. TUE



Greater than 1 eigen value indicates that the principal component captures more information than any individual variable in the dataset.





Based on the correlation values and the principal component analysis the most correlated and most contributing attributes are following**:**

- Gender,
- Height,
- Age,
- FCVC
- CH20
- CALC
- TUE

We can remove other attributes

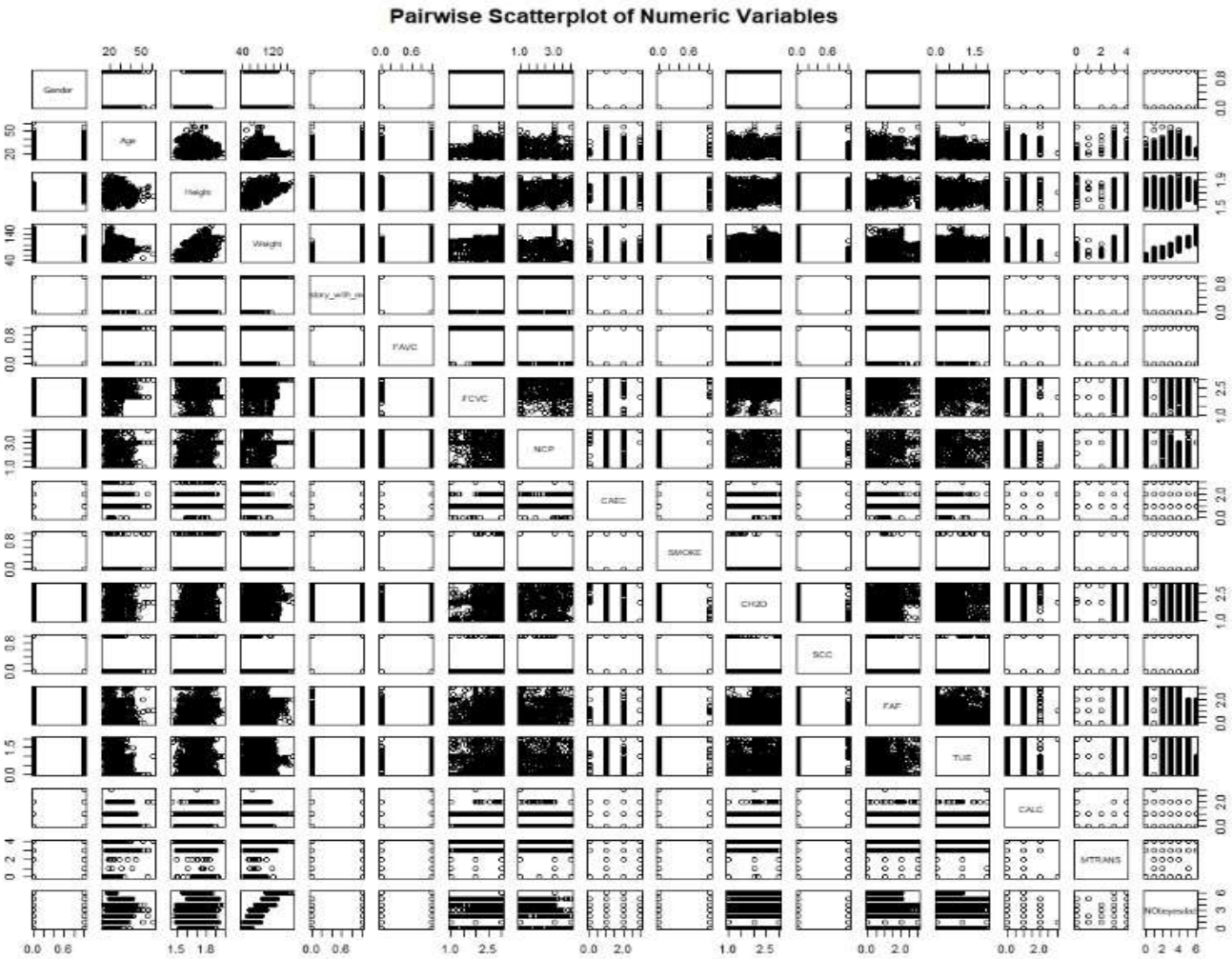
```
elim_subset_data <- subset(obesity_encoded, select = c("Height","Gender","CH2O", "CALC", "FCVC", "Age", "TUE"))
elim_subset_data
> fviz_eig(pca_res, addlabels = TRUE, ncp = 15, barfill = "beige", barcolor = "lightblue", linecolor = "lightgreen")
> elim_subset_data
   Height Gender CH2O CALC FCVC Age TUE
1    1.62      1    2    0    2  21   1
2    1.52      1    3    1    3  21   0
3    1.80      0    2    2    2  23   1
4    1.80      0    2    2    3  27   0
5    1.78      0    2    1    2  22   0
6    1.62      0    2    1    2  29   0
7    1.50      1    2    1    3  23   0
8    1.64      0    2    1    2  22   0
9    1.78      0    2    2    3  24   1
10   1.72      0    2    0    2  22   1
11   1.85      0    3    1    3  26   2
12   1.72      1    2    1    2  21   1
13   1.65      0    3    1    3  22   0
14   1.80      0    2    2    2  41   1
15   1.77      0    1    1    3  23   1
16   1.70      1    2    1    3  22   1
17   1.93      0    1    1    2  27   0
18   1.53      1    2    0    2  29   0
19   1.71      1    1    0    3  30   0
20   1.65      1    2    1    2  23   0
21   1.65      0    2    0    2  22   2
22   1.69      1    2    0    3  52   0
23   1.65      1    2    1    3  22   0
24   1.60      1    2    1    1  22   2
25   1.85      0    2    1    2  21   1
```

2c) You may need to translate alphanumeric (e.g., character) values into numeric values. Create a mapping for each field of values to integers. Make sure you put these mapping tables into your report.

```
directory <- "C:/Users/rajit/Downloads"
obesity_data_file <- file.path(directory, "ObesityDataSet_raw_and_data_sinthetic.csv")
ObesityDataSet <- read.table(file = obesity_data_file, stringsAsFactors = FALSE, sep = ',', header = TRUE)

ObesityDataSet_num <- ObesityDataSet

ObesityDataSet_num <- ObesityDataSet_num %>%
  mutate(
    CAEC = ifelse(CAEC == "no", 0,
                  ifelse(CAEC == "Sometimes", 1,
                         ifelse(CAEC == "Frequently", 2, 3))),
    CALC = ifelse(CALC == "no", 0,
                  ifelse(CALC == "Sometimes", 1,
                         ifelse(CALC == "Frequently", 2, 3))),
    MTRANS = ifelse(MTRANS == "Automobile", 3,
                    ifelse(MTRANS == "Bike", 1,
                           ifelse(MTRANS == "Motorbike", 2,
                                  ifelse(MTRANS == "Public_Trans", 4,0)))),
    NObeyesdad = ifelse(NObeyesdad == "Insufficient_Weight", 0,
                        ifelse(NObeyesdad == "Normal_Weight", 1,
                               ifelse(NObeyesdad == "Overweight_Level_I", 2,
                                      ifelse(NObeyesdad == "Overweight_Level_II", 3,
                                             ifelse(NObeyesdad == "Obesity_Type_I", 4,
                                                    ifelse(NObeyesdad == "Obesity_Type_II", 5, 6))))),
    Gender = ifelse(Gender == "Male", 1, 0),
    family_history_with_overweight = ifelse(family_history_with_overweight == "no", 0, 1),
    FAVC = ifelse(FAVC == "no", 0, 1),
    SMOKE = ifelse(SMOKE == "no", 0, 1),
    SCC = ifelse(SCC == "no", 0, 1)
  )
```

Column Name	Original Values	Encoded Values	Description
family_history_with_overweight	yes, no	1, 0	yes = 1, no = 0
FAVC	yes, no	1, 0	yes = 1, no = 0
SMOKE	yes, no	1, 0	yes = 1, no = 0
SCC	yes, no	1, 0	yes = 1, no = 0
Gender	Male, Female	1, 0	Male = 1, Female = 0
CALC	no, Sometimes, Frequently, Always	0, 1, 2, 3,	no = 0, Sometimes = 1, Frequently = 2, Always = 3
CAEC	no, Sometimes, Frequently, Always	0, 1, 2, 3,	no = 0, Sometimes = 1, Frequently = 2, Always = 3

NObeyesdad	Insufficient_Weight, Normal_Weight, Overweight_Level_I, Overweight_Level_II, Obesity_Type_I, Obesity_Type_II, Obesity_Type_III	0, 1, 2, 3, 4, 5, 6	Insufficient_Weight = 0, Normal_Weight = 1, Overweight_Level_I = 2, Overweight_Level_II = 3, Obesity_Type_I = 4, Obesity_Type_II = 5, Obesity_Type_III = 6
MTRANS	Walking, Bike, Motorbike, Automobile, Public_Transportation	0, 1, 2, 3, 4	Walking = 0, Bike = 1, Motorbike = 2, Automobile = 3, Public_Transportation = 4



2d) Split the original data set into Training and Test Sets. Use 70-30%, 60-40%, 50-50%

```r
subset_data_70_30 <- obesity_encoded
subset_data_70_30.rowsnum = nrow(subset_data_70_30)
subset_data_70_30.sample = 0.7
subset_data_70_30.rows = subset_data_70_30.sample * subset_data_70_30.rowsnum
subset_data_70_30.rows
subset_data_70_30.train.index = sample(subset_data_70_30.rowsnum,subset_data_70_30.rows)
length(subset_data_70_30.train.index)
subset_data_70_30.train = subset_data_70_30[subset_data_70_30.train.index,]
subset_data_70_30.train[1:20,]
subset_data_70_30.test = subset_data_70_30[-subset_data_70_30.train.index,]
subset_data_70_30.test[1:20,]


subset_data_60_40 <- obesity_encoded
subset_data_60_40.rowsnum = nrow(subset_data_60_40)
subset_data_60_40.sample = 0.6
subset_data_60_40.rows = subset_data_60_40.sample * subset_data_60_40.rowsnum
subset_data_60_40.rows
subset_data_60_40.train.index = sample(subset_data_60_40.rowsnum,subset_data_60_40.rows)
length(subset_data_60_40.train.index)
subset_data_60_40.train = subset_data_60_40[subset_data_60_40.train.index,]
subset_data_60_40.train[1:20,]
subset_data_60_40.test = subset_data_60_40[-subset_data_60_40.train.index,]
subset_data_60_40.test[1:20,]


subset_data_50_50 <- obesity_encoded
subset_data_50_50.rowsnum = nrow(subset_data_50_50)
subset_data_50_50.sample = 0.5
subset_data_50_50.rows = subset_data_50_50.sample * subset_data_50_50.rowsnum
subset_data_50_50.rows
subset_data_50_50.train.index = sample(subset_data_50_50.rowsnum,subset_data_50_50.rows)
length(subset_data_50_50.train.index)
subset_data_50_50.train = subset_data_50_50[subset_data_50_50.train.index,]
subset_data_50_50.train[1:20,]
subset_data_50_50.test = subset_data_50_50[-subset_data_50_50.train.index,]
subset_data_50_50.test[1:20,]
```

267:1 (Top Level)

Console Terminal Background Jobs

R 4.3.3 · ~/

```
36  21   1.50   65.0                    1   0   2   1   2   2   1    4    3
> length(subset_data_70_30.train.index)
[1] 1477
> length(subset_data_60_40.train.index)
[1] 1266
> length(subset_data_50_50.train.index)
[1] 1055
> subset_data_70_30.rows
[1] 1477.7
> subset_data_60_40.rows
[1] 1266.6
> subset_data_50_50.rows
[1] 1055.5
>
```

NORMALIZATION

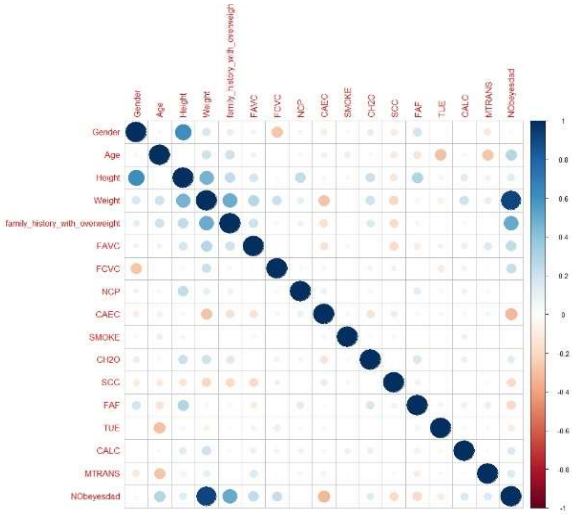
Compare the normalization results of min-max vs. z-score normalization.

```r
#----------------------------------NORMALIZATION-----------------------------------------------------------------
#Min-Max Normalization
normalize <- function(x) {((x - min(x)) / (max(x) - min(x)))}
normalize
obesity_encoded.norm <- as.data.frame(lapply(obesity_encoded,normalize))
obesity_encoded.norm[1:17]

#Z-score normalization:

# Store the means and standard deviations before normalization
feature_means <- apply(obesity_encoded, 2, mean)
feature_stds <- apply(obesity_encoded, 2, sd)


zscore<- function(x){(x-mean(x))/sd(x)}
obesity_encoded.znorm<- as.data.frame(lapply(obesity_encoded,scale))
obesity_encoded.znorm[1:17]

# Function to retrieve original values from Z-score normalized values
retrieve_original <- function(znorm_value, mean_value, sd_value) {
  original_value <- (znorm_value * sd_value) + mean_value
  return(original_value)
}
retrieved_original_values <- as.data.frame(mapply(retrieve_original,
                                                  obesity_encoded.znorm,
                                                  feature_means,
                                                  feature_stds))

# Compare the original and retrieved values returns TRUE if the original and retrieved values are the same
all.equal(obesity_encoded, retrieved_original_values)
obesity_encoded
retrieved_original_values

#-----------------------------------------------------------------------------------------------------------------
```

MIN-MAX NORMALIZATION:

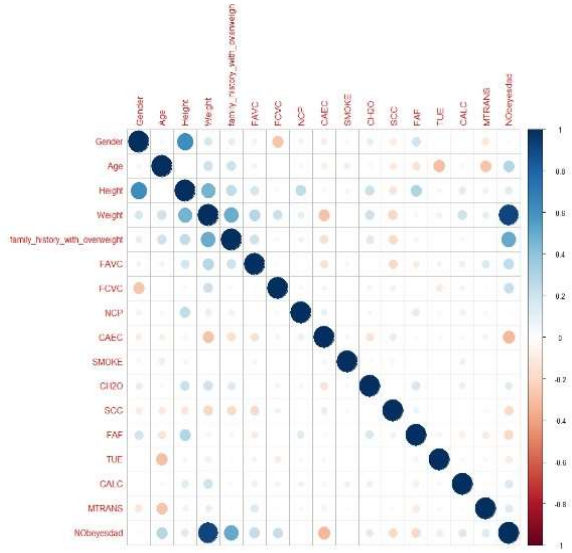
```
R 4.3.3 · ~/
> #----------------------------------NORMALIZATION-----------------------------------------------------------------
> #Min-Max Normalization
> normalize <- function(x) {((x - min(x)) / (max(x) - min(x)))}
> normalize
function(x) {((x - min(x)) / (max(x) - min(x)))}
> obesity_encoded.norm <- as.data.frame(lapply(obesity_encoded,normalize))
> obesity_encoded.norm[1:17]
   Gender       Age      Height     Weight family_history_with_overweight FAVC FCVC       NCP      CAEC SMOKE CH2O SCC       FAF TUE      CALC MTRANS NObeyesdad
1       0 0.1489362 0.32075472 0.18656716                             1    0  0.5 0.6666667 0.3333333     0  0.5   0 0.0000000 0.5 0.0000000   1.00  0.1666667
2       0 0.1489362 0.13207547 0.12686567                             1    0  1.0 0.6666667 0.3333333     1  1.0   1 1.0000000 0.0 0.3333333   1.00  0.1666667
3       1 0.1914894 0.66037736 0.28358209                             1    0  0.5 0.6666667 0.3333333     0  0.5   0 0.6666667 0.5 0.6666667   1.00  0.1666667
4       1 0.2765957 0.66037736 0.35820896                             0    0  1.0 0.6666667 0.3333333     0  0.5   0 0.6666667 0.0 0.6666667   0.00  0.3333333
5       1 0.1702128 0.62264151 0.37910448                             0    0  0.5 0.0000000 0.3333333     0  0.5   0 0.0000000 0.0 0.3333333   1.00  0.5000000
6       1 0.3191489 0.32075472 0.10447761                             0    1  0.5 0.6666667 0.3333333     0  0.5   0 0.0000000 0.0 0.3333333   0.75  0.1666667
7       0 0.1914894 0.09433962 0.11940299                             1    1  1.0 0.6666667 0.3333333     0  0.5   0 0.3333333 0.0 0.3333333   0.50  0.1666667
8       1 0.1702128 0.35849057 0.10447761                             0    0  0.5 0.6666667 0.3333333     0  0.5   0 1.0000000 0.0 0.3333333   1.00  0.1666667
9       1 0.2127660 0.62264151 0.18656716                             1    1  1.0 0.6666667 0.3333333     0  0.5   0 0.3333333 0.5 0.6666667   1.00  0.1666667
10      1 0.1702128 0.50943396 0.21641791                             1    1  0.5 0.6666667 0.3333333     0  0.5   0 0.3333333 0.5 0.0000000   1.00  0.1666667
11      1 0.2553191 0.75471698 0.49253731                             1    1  1.0 0.6666667 0.6666667     0  1.0   0 0.6666667 1.0 0.3333333   1.00  0.6666667
12      0 0.1489362 0.50943396 0.30597015                             1    1  0.5 0.6666667 0.6666667     0  0.5   1 0.6666667 0.5 0.3333333   1.00  0.5000000
13      1 0.1702128 0.37735849 0.12686567                             0    0  1.0 0.6666667 0.3333333     0  1.0   0 0.6666667 0.0 0.3333333   1.00  0.1666667
14      1 0.5744681 0.66037736 0.44776119                             0    1  0.5 0.6666667 0.3333333     0  0.5   0 0.6666667 0.5 0.6666667   0.75  0.6666667
15      1 0.1914894 0.60377358 0.15671642                             1    1  1.0 0.0000000 0.3333333     0  0.0   0 0.3333333 0.5 0.3333333   1.00  0.1666667
16      0 0.1702128 0.47169811 0.20149254                             1    0  1.0 0.6666667 1.0000000     0  0.5   1 0.6666667 0.5 0.3333333   1.00  0.1666667
17      1 0.2765957 0.90566038 0.47014925                             1    1  0.5 0.0000000 0.3333333     0  0.0   0 0.3333333 0.0 0.3333333   1.00  0.5000000
18      0 0.3191489 0.15094340 0.29104478                             0    1  0.5 0.0000000 0.3333333     0  0.5   0 0.0000000 0.0 0.0000000   0.75  0.6666667
19      0 0.3404255 0.49056604 0.32089552                             1    1  1.0 1.0000000 0.6666667     1  0.0   0 0.0000000 0.0 0.0000000   0.75  0.5000000
20      0 0.1914894 0.37735849 0.23134328                             1    0  0.5 0.0000000 0.3333333     0  0.5   0 0.0000000 0.0 0.3333333   1.00  0.3333333
21      1 0.1702128 0.37735849 0.30597015                             1    0  0.5 0.6666667 0.3333333     0  0.5   0 1.0000000 1.0 0.0000000   0.00  0.5000000
22      0 0.8085106 0.45283019 0.35820896                             1    1  1.0 0.0000000 0.3333333     1  0.5   0 0.0000000 0.0 0.0000000   0.75  0.6666667
23      0 0.1702128 0.37735849 0.15671642                             1    1  1.0 0.6666667 0.3333333     0  0.5   0 0.3333333 0.0 0.3333333   0.75  0.1666667
24      0 0.1702128 0.28301887 0.32089552                             1    1  0.0 0.0000000 0.3333333     0  0.5   0 0.0000000 1.0 0.3333333   1.00  0.6666667
25      1 0.1489362 0.75471698 0.21641791                             1    1  0.5 0.6666667 0.3333333     0  0.5   0 0.0000000 0.5 0.3333333   1.00  0.1666667
26      1 0.1276596 0.28301887 0.08208955                             1    0  0.5 1.0000000 0.6666667     1  0.5   0 1.0000000 1.0 0.0000000   1.00  0.1666667
27      1 0.1489362 0.47169811 0.19402985                             1    1  0.5 0.0000000 0.6666667     0  0.5   0 0.3333333 1.0 1.0000000   0.00  0.1666667
28      0 0.1914894 0.28301887 0.09701493                             0    1  0.5 1.0000000 0.6666667     0  0.5   0 0.6666667 0.5 0.3333333   0.75  0.1666667
29      1 0.1063830 0.56603774 0.27611940                             1    1  1.0 0.6666667 0.3333333     0  0.5   1 1.0000000 0.5 0.3333333   1.00  0.1666667
30      1 0.1914894 0.43396226 0.23134328                             0    1  0.5 0.6666667 0.3333333     0  0.5   0 0.6666667 1.0 0.6666667   0.00  0.1666667
```

```r
> obesity_encoded_norm_corr <- cor(obesity_encoded.norm)
> corrplot::corrplot(obesity_encoded_norm_corr)
>
```



Z-SCORE NORMALIZATION:

```
> zscore<- function(x){(x-mean(x))/sd(x)}
> obesity_encoded.znorm<- as.data.frame(lapply(obesity_encoded,scale))
> obesity_encoded.znorm[1:17]
       Gender        Age     Height     Weight family_history_with_overweight       FAVC       FCVC       NCP      CAEC     SMOKE       CH2O        SCC        FAF        TUE       CALC     MTRANS  NObeyesdad
1  -1.0116740 -0.52200070 -0.87538193 -0.86235386                   0.4721794 -2.7591155 -0.7848327  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203 -1.1877577  0.5618636 -1.4188354  0.4525799 -1.06408228
2  -1.0116740 -0.52200070 -1.94713793 -1.16780030                   0.4721794 -2.7591155  1.0880839  0.404057 -0.3002744  6.8523733  1.61837509  4.5803537  2.3391959 -1.0803687  0.5210361  0.4525799 -1.06408228
3   0.9879925 -0.20683997  1.05377886 -0.36600341                   0.4721794 -2.7591155 -0.7848327  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203  1.1635447  0.5618636  2.4609075  0.4525799 -1.06408228
4   0.9879925  0.42348149  1.05377886  0.01580463                  -2.1168357 -2.7591155  1.0880839  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203  1.1635447 -1.0803687  2.4609075 -4.8258519 -0.56031962
5   0.9879925 -0.36442034  0.83942766  0.12271089                  -2.1168357 -2.7591155 -0.7848327 -2.166509 -0.3002744 -0.1458657 -0.01307017 -0.2182203 -1.1877577 -1.0803687  0.5210361  0.4525799 -0.05655696
6   0.9879925  0.73864222 -0.87538193 -1.28234271                  -2.1168357  0.3622633 -0.7848327  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203 -1.1877577 -1.0803687  0.5210361 -0.8670281 -1.06408228
7  -1.0116740 -0.20683997 -2.16148912 -1.20598110                   0.4721794  0.3622633  1.0880839  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203 -0.0121065 -1.0803687  0.5210361 -2.1866360 -1.06408228
8   0.9879925 -0.36442034 -0.66103073 -1.28234271                  -2.1168357 -2.7591155 -0.7848327  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203  2.3391959 -1.0803687  0.5210361  0.4525799 -1.06408228
9   0.9879925 -0.04925961  0.83942766 -0.86235386                   0.4721794  0.3622633  1.0880839  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203 -0.0121065  0.5618636  2.4609075  0.4525799 -1.06408228
10  0.9879925 -0.36442034  0.19637406 -0.70963065                   0.4721794  0.3622633 -0.7848327  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203 -0.0121065  0.5618636 -1.4188354  0.4525799 -1.06408228
11  0.9879925  0.26590112  1.58965686  0.70305911                   0.4721794  0.3622633  1.0880839  0.404057  1.8339993 -0.1458657  1.61837509 -0.2182203  1.1635447  2.2040959  0.5210361  0.4525799  0.44720570
12 -1.0116740 -0.52200070  0.19637406 -0.25146100                   0.4721794  0.3622633 -0.7848327  0.404057  1.8339993 -0.1458657 -0.01307017  4.5803537  1.1635447  0.5618636  0.5210361  0.4525799 -0.05655696
13  0.9879925 -0.36442034 -0.55385513 -1.16780030                  -2.1168357 -2.7591155  1.0880839  0.404057 -0.3002744 -0.1458657  1.61837509 -0.2182203  1.1635447 -1.0803687  0.5210361  0.4525799 -1.06408228
14  0.9879925  2.62960661  1.05377886  0.47397428                  -2.1168357  0.3622633 -0.7848327  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203  1.1635447  0.5618636  2.4609075 -0.8670281  0.44720570
15  0.9879925 -0.20683997  0.73225206 -1.01507708                   0.4721794  0.3622633  1.0880839 -2.166509 -0.3002744 -0.1458657 -1.64451542 -0.2182203 -0.0121065  0.5618636  0.5210361  0.4525799 -1.06408228
16 -1.0116740 -0.36442034 -0.01797714 -0.78599225                   0.4721794 -2.7591155  1.0880839  0.404057  3.9682730 -0.1458657 -0.01307017  4.5803537  1.1635447  0.5618636  0.5210361  0.4525799 -1.06408228
17  0.9879925  0.42348149  2.44706165  0.58851670                   0.4721794  0.3622633 -0.7848327 -2.166509 -0.3002744 -0.1458657 -1.64451542 -0.2182203 -0.0121065 -1.0803687  0.5210361  0.4525799 -0.05655696
18 -1.0116740  0.73864222 -1.83996233 -0.32782260                  -2.1168357  0.3622633 -0.7848327 -2.166509 -0.3002744 -0.1458657 -0.01307017 -0.2182203 -1.1877577 -1.0803687 -1.4188354 -0.8670281  0.44720570
19 -1.0116740  0.89622259  0.08919846 -0.17509939                   0.4721794  0.3622633  1.0880839  1.689340  1.8339993  6.8523733 -1.64451542 -0.2182203 -1.1877577 -1.0803687 -1.4188354 -0.8670281 -0.05655696
20 -1.0116740 -0.20683997 -0.55385513 -0.63326904                   0.4721794 -2.7591155 -0.7848327 -2.166509 -0.3002744 -0.1458657 -0.01307017 -0.2182203 -1.1877577 -1.0803687  0.5210361  0.4525799 -0.56031962
21  0.9879925 -0.36442034 -0.55385513 -0.25146100                   0.4721794 -2.7591155 -0.7848327  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203  2.3391959  2.2040959 -1.4188354 -4.8258519 -0.05655696
22 -1.0116740  4.36299063 -0.12515273  0.01580463                   0.4721794  0.3622633  1.0880839 -2.166509 -0.3002744  6.8523733 -0.01307017 -0.2182203 -1.1877577 -1.0803687 -1.4188354 -0.8670281  0.44720570
23 -1.0116740 -0.36442034 -0.55385513 -1.01507708                   0.4721794  0.3622633  1.0880839  0.404057 -0.3002744 -0.1458657 -0.01307017 -0.2182203 -0.0121065 -1.0803687  0.5210361 -0.8670281 -1.06408228
24 -1.0116740 -0.36442034 -1.08973313 -0.17509939                   0.4721794  0.3622633 -2.6577494 -2.166509 -0.3002744 -0.1458657 -0.01307017 -0.2182203 -1.1877577  2.2040959  0.5210361  0.4525799  0.44720570
```



Retrieve original values from Z-score normalized values:

```
# Function to retrieve original values from Z-score normalized values
retrieve_original <- function(znorm_value, mean_value, sd_value) {
  original_value <- (znorm_value * sd_value) + mean_value
  return(original_value)
}
retrieved_original_values <- as.data.frame(mapply(retrieve_original,
                                                  obesity_encoded.znorm,
                                                  feature_means,
                                                  feature_stds))

# Compare the original and retrieved values returns TRUE if the original and retrieved values are the same
all.equal(obesity_encoded, retrieved_original_values)
obesity_encoded
retrieved_original_values
```

```
> # Function to retrieve original values from Z-score normalized values
> retrieve_original <- function(znorm_value, mean_value, sd_value) {
+   original_value <- (znorm_value * sd_value) + mean_value
+   return(original_value)
+ }
> retrieved_original_values <- as.data.frame(mapply(retrieve_original,
+                                                   obesity_encoded.znorm,
+                                                   feature_means,
+                                                   feature_stds))
> # Compare the original and retrieved values returns TRUE if the original and retrieved values are the same
> all.equal(obesity_encoded, retrieved_original_values)
[1] TRUE
```

3. CLUSTERING ON THE WHOLE DATA SET

3a) Use the whole date set when clustering (the first part of the rubric).

The below screenshots show the cluster means of each feature that we selected for modeling and then its cluster vector. Finally, between_ss /total_ss tells us how much of the variability in the data is accounted for by the clusters found by the algorithm. We used the factoextra package which provides some easy-to-use functions to extract and visualize the output of multivariate data analyses and fviz _cluster is ggplot2-based elegant visualization of partitioning methods including k-means. It takes the k means dataset and normalizes the dataset to represent output.

2

```
K-means clustering with 2 clusters of sizes 825, 1286

Cluster means:
     Gender      Age   Height    Weight family_history_with_overweight       FAVC      FCVC
1 0.6012121 25.46517 1.743277 114.24491                     0.9939394 0.9818182 2.569646
2 0.4447900 23.57320 1.674990  68.84224                     0.7045101 0.8211509 2.322428
      NCP     CAEC      SMOKE      CH2O       SCC      FAF      TUE     CALC   MTRANS
1 2.805668 1.021818 0.02303030 2.109198 0.002424242 0.9155039 0.6425882 0.8375758 3.790303
2 2.608619 1.216952 0.01944012 1.943098 0.073094868 1.0711102 0.6676669 0.6632970 3.571540
  NObeyesdad  cluster
1   5.103030 2.873939
2   1.835148 1.517885

Clustering vector:
   [1] 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
  [44] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
  [87] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
 [130] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2
 [173] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2
 [216] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1
 [259] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2
 [302] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
 [345] 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2
 [388] 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2
 [431] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
 [474] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
 [517] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [560] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [603] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

3

```
K-means clustering with 3 clusters of sizes 620, 770, 721

Cluster means:
     Gender      Age   Height    Weight family_history_with_overweight       FAVC      FCVC      NCP
1 0.3241935 20.89596 1.651587  56.03956                     0.5258065 0.7677419 2.369540 2.742440
2 0.6103896 26.01498 1.754650  82.87181                     0.8844156 0.8831182 2.232218 2.504297
3 0.5506241 25.43255 1.741576 116.82040                     0.9972261 0.9861304 2.639774 2.830428
      CAEC      SMOKE      CH2O       SCC      FAF      TUE     CALC   MTRANS NObeyesdad Cluster
1 1.359677 0.02096774 1.861485 0.120967742 1.114029 0.7768629 0.6096774 3.627419  0.8016129       1
2 1.083117 0.01818182 2.033905 0.024675325 1.071351 0.6194402 0.6896104 3.529870  7.9298701       2
3 1.013870 0.02557836 2.106359 0.002773925 0.855895 0.5965756 0.8807212 3.818308  5.2940361       3

Clustering vector:
   [1] 1 1 2 2 2 1 1 1 1 1 3 2 1 2 1 1 3 2 2 2 2 2 1 2 1 1 1 1 2 2 2 2 2 2 1 1 1 1 2 2 1 2
  [49] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 3 1 2 1 1 1 2 1 1 2 2 1 1 2 2 1 2 2 1 2 2 1 2
  [97] 1 1 1 2 1 1 1 1 2 1 2 1 2 2 2 2 1 1 1 1 1 2 2 1 2 2 2 1 1 2 2 1 1 1 2 1 1 2 3 2 2 2
 [145] 1 2 1 2 1 2 2 1 1 2 2 1 1 1 2 2 2 1 2 2 3 2 2 2 2 2 1 1 2 1 1 2 2 1 1 1 2 1 2 2 3 1
 [193] 2 1 2 2 2 1 1 1 2 2 1 2 2 2 1 2 1 1 3 1 1 1 1 1 2 2 1 2 1 1 2 2 1 1 1 2 3 1 1 2 3 1
 [241] 1 1 1 2 1 1 1 1 2 2 2 2 2 1 1 2 2 3 2 2 2 1 1 1 1 1 2 1 1 3 1 1 2 1 1 1 1 1 1 1 1 3
 [289] 1 1 2 2 1 1 2 3 1 1 2 2 2 2 1 1 1 2 3 1 1 2 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 2 2 1 1 2
 [337] 1 1 1 1 2 1 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2 1 2 2 1 1 1 1 2 1 1 2
 [385] 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 2 1 1 1 2 2 1 2 2 2 2 2 1 1 1 2
 [433] 1 1 2 2 1 1 1 1 1 1 1 1 1 1 2 2 3 1 2 1 1 1 1 1 1 1 1 2 1 3 2 1 2 1 2 2 1 1 1 2 2 1
 [481] 1 1 1 1 2 2 2 3 2 1 2 2 1 2 1 1 1 1 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [529] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [577] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [625] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [673] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [721] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2
 [769] 2 2 1 2 2 1 2 2 2 2 2 2 2 2 1 2 2 1 1 1 2 2 2 1 1 1 1 1 2 2 2 2 2 1 2 1 2 2 2 2 2 1
 [817] 2 2 2 2 2 2 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1
 [865] 2 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1
 [913] 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 1 1 1 1 2 1 1 1
 [961] 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 1 2 2 2 2 2 2
 [ reached getOption("max.print") -- omitted 1111 entries ]

Within cluster sum of squares by cluster:
[1]  58493.35  93451.43 131429.08
 (between_SS / total_SS =  81.7 %)
```

4

```
> obesity_cluster.k4
K-means clustering with 4 clusters of sizes 640, 703, 212, 556

Cluster means:
    Gender      Age   Height   Weight family_history_with_overweight      FAVC      FCVC      NCP
1 0.5606375 20.54585 1.652808 56.62701                      0.5016201 0.7701316 2.355064 2.660506
2 0.5405405 25.45928 1.741852 117.2432                      0.9751550 0.9857752 2.637958 2.857272
3 0.5377358 37.13962 1.663153 79.25612                      0.8915057 0.9150566 2.572975 2.750674
4 0.6978417 22.31142 1.719148  85.10334                      0.9262590 0.8884892 2.239447 2.544256
      CAEC     SMOKE     CH2O      SCC      FAF      TUE     CALC     MTRANS Nobeyesdad  cluster
1 1.317187 0.02187500 1.884677 0.11718750 1.1086504 0.7592475 0.6218750 3.651562  0.8296875 1.053125
2 1.317057 0.02443808 2.125979 0.00284495 0.6538314 0.5820710 0.8663841 3.811005  0.5886775 1.000000
3 1.103774 0.01886792 1.877431 0.02830189 0.8623710 0.2991870 0.6792453 2.966981  2.9858491 1.938679
4 1.113309 0.01618705 2.050612 0.02338129 1.1487951 0.7737635 0.6654676 3.726619  3.0107914 2.030576

Clustering vector:
  [1] 1 1 4 4 1 1 1 1 1 2 4 1 3 1 1 2 3 1 4 3 1 4 1 1 1 4 4 3 4 3 1 1 1 1 4 4 1 1 4 1 4 4
 [29] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [57] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [85] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[113] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[141] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[169] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[197] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[225] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[253] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[281] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[309] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[337] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[365] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[393] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[421] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[449] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[477] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[505] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[533] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[561] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[589] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[617] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[645] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[673] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[701] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[729] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[757] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[785] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[813] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[841] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[869] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[897] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[925] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[953] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[981] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[1000] 1

Within cluster sum of squares by cluster:
[1] 60438.12 125836.77  19752.03  41136.33
 (between_SS / total_SS =  84.1 %)
```

5

```
K-means clustering with 5 clusters of sizes 361, 669, 506, 182, 393

Cluster means:
    Gender      Age   Height   Weight family_history_with_overweight      FAVC      FCVC      NCP
1 0.2520776 20.13102 1.646628  50.12066                      0.4182825 0.7922438 2.374145 2.796864
2 0.5381166 25.34780 1.741626 118.0175                      0.9970105 0.9850323 2.635766 2.877294
3 0.6852708 27.81494 1.719100  87.91615                      0.9387152 0.9278249 2.738883 2.465252
4 0.3626374 21.90041 1.698527  81.19790                      0.8956044 0.8956044 2.378847 2.504895
5 0.5190840 22.06814 1.668650  67.26694                      0.6870229 0.7404580 2.344300 2.649834
      CAEC     SMOKE     CH2O      SCC      FAF      TUE     CALC     MTRANS Nobeyesdad  cluster
1 1.468144 0.01385042 1.794727 0.102493075 1.1492192 0.8417946 0.6121884 3.742382  0.3462604 1.000000
2 1.013453 0.02391629 2.167974 0.002988537 0.8688561 0.5816191 0.8073248 3.819133  5.3243647 5.000000
3 1.083004 0.01383390 2.005715 0.017786561 1.1049979 0.7610891 0.6284583 3.741107  3.3221344 2.098814
4 1.093407 0.02197802 1.856621 0.021978022 0.8383927 0.3027735 0.6318681 2.989011  3.1923077 2.010989
5 1.152672 0.03051435 2.013205 0.111959788 1.0859603 0.6502497 0.7201018 3.503817  1.5801527 1.340967

Clustering vector:
  [1] 1 1 1 1 1 1 5 5 2 3 1 4 5 5 3 4 4 5 3 4 5 3 5 1 5 1 5 1 5 5 3 5 5 4 5 5 1 1 5 5 3 5 5 5 3 5 5 5
 [29] 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [57] 5 1 1 1 5 5 5 5 1 1 4 3 1 3 3 3 5 3 3 1 5 3 3 3 5 3 3 3 3 3 1 5 5 5 4 5 5 5 3 3 1 1 4 2 5 5 4 5
 [85] 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[113] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[141] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[169] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[197] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[225] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[253] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[281] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[309] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[337] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[365] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[393] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[421] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[449] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[477] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[505] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[533] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[561] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[589] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[617] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[645] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[673] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[701] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[729] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[757] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[785] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[813] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[841] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[869] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[897] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[925] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[953] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[981] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[1000] 1

Within cluster sum of squares by cluster:
[1] 8433.560 14102.869 11071.426 13435.978 11648.144  8834.667  7208.983
 (between_SS / total_SS =  94.2 %)
```

6

```
Cluster means:
    Gender      Age   Height   Weight family_history_with_overweight      FAVC      FCVC      NCP
1 0.5483333 26.63179 1.701739  81.96268                      0.9083333 0.8950000 2.276609 2.505481
2 0.4832402 21.75721 1.662681  65.13138                      0.6452514 0.7318436 2.345555 2.620479
3 0.3125828 21.62098 1.770910 136.5751                      1.0000000 0.9933775 2.928813 3.000000
4 0.7241379 25.32618 1.719524 101.97007                      0.9874608 0.9780564 2.488439 2.574605
5 0.6923077 27.07608 1.751691 116.04829                      0.9971510 0.9800570 2.495271 2.940210
6 0.4228402 20.20548 1.642104  49.41045                      0.4036145 0.7861446 2.376711 2.775990
      CAEC     SMOKE     CH2O      SCC      FAF      TUE     CALC     MTRANS Nobeyesdad  cluster
1 1.1050000 0.01666667 1.990968 0.03000000 1.0035434 0.5779608 0.6933333 3.505000  2.908333 2.000000
2 1.142458 0.02793296 2.007240 0.111731844 1.1255260 0.6402621 0.6368715 3.458101  1.4497207 1.195531
3 1.006623 0.01987510 2.159918 0.000000000 0.1432386 0.7588998 1.0000000 3.993377  5.8940397 3.000000
4 1.031348 0.01253918 1.935408 0.000000000 0.9200300 0.7875841 0.6206897 3.802508  4.5987461 2.686520
5 1.017094 0.02564103 2.243960 0.005698006 0.6896332 0.4650989 0.9629630 3.703704  5.2478632 1.000000
6 1.500000 0.01506024 1.790860 0.108433735 1.1320485 0.8544768 0.6415663 3.804217  0.3222892 1.000000

Clustering vector:
  [1] 2 6 1 1 1 6 6 6 2 2 4 1 6 4 2 2 4 1 1 2 1 1 2 1 2 6 2 6 1 2 1 2 1 1 1 2 2 6 2 1 1 2 2 2 1 2 2 2
 [49] 2 6 6 6 6 6 6 6 2 2 2 6 6 2 2 6 6 2 1 4 5 2 1 6 2 2 1 6 1 1 2 2 1 1 6 1 2 1 1 2 1 4 2 1 6 6 2
 [97] 2 6 6 2 2 2 6 2 1 6 6 1 4 1 1 2 6 6 2 6 1 1 6 1 4 6 2 1 1 2 2 2 6 1 3 2 6 1 5 1 2 1 1 2 2 2 4 1
[145] 2 6 1 2 1 1 1 2 1 1 4 6 2 2 4 1 2 1 2 1 1 1 1 2 2 2 6 1 3 2 6 1 5 2 2 4 4 4 2 6 1
[193] 1 2 1 1 2 5 6 2 1 1 4 1 2 1 4 1 2 2 5 6 2 6 2 2 6 1 1 2 2 1 5 6 2 1 5 1 2 6 1 6 2 2 6 6 2
[241] 2 2 2 4 6 6 2 6 1 1 1 6 6 1 1 2 1 1 2 6 6 1 2 2 6 2 3 6 2 2 2 6 2 2 6 2 2 4 2 2
[289] 6 1 2 2 2 2 4 2 6 1 1 1 2 6 6 6 1 5 2 6 1 2 6 2 6 2 2 1 2 6 2 2 2 6 1 2 1 3 6 1 6 6 6 6
[337] 6 2 1 3 2 6 6 4 2 1 6 6 4 1 5 2 2 6 6 2 2 1 1 2 2 1 1 2 2 2 2 6 2 1 1 2 2 2 1 1 6 4
[385] 2 6 6 1 4 2 6 6 2 1 6 6 2 2 5 2 2 1 4 5 1 1 2 2 1 6 2 2 1 1 5 2 1 1 1 1 2 2 2 4 1 6 2 2 1 1 6
[433] 2 2 1 1 6 6 2 2 6 2 6 2 6 1 4 2 2 2 2 6 2 2 6 1 6 4 1 2 6 1 1 6 6 6 1 2 6 2 2 6 2 6
[481] 6 2 6 2 1 2 1 4 1 6 1 2 2 2 2 6 6 4 3 3 3 5 4 3 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
[529] 6 6 6 2 2 2 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 2 6 6 6 2 2 6 6 6 6 6 6 6 6 6 6 6 6 6
[577] 6 6 6 2 2 6 6 6 6 6 6 6 6 6 6 6 2 2 2 2 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
[625] 6 6 2 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
[673] 6 2 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
[721] 6 6 6 6 6 6 6 6 2 2 2 2 6 6 6 6 6 6 6 6 6 6 6 6 1 2 1 1 2 1 1 1 1 1 1 1 2 2 2 2 2 1 2 2 2 1
[769] 1 1 1 1 2 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1
[817] 1 1 1 1 1 1 2 2 2 2 2 1 1 2 2 2 2 2 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2
[865] 1 6 6 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 1 6 1 2 2 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1
[913] 2 2 2 2 2 2 1 1 1 1 2 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 6 6 2 1 1 2 2 2 2
[961] 2 2 2 2 2 1 1 1 1 1 1 1 2 1 2 1 1 1 1 4 1 1 1 2 1 1 2 1 2 2 1 2 2 1 2 1 1 1 1 1 1 1
[ reached getOption("max.print") -- omitted 1111 entries ]

Within cluster sum of squares by cluster:
[1] 56904.41 18209.54 13466.50 15363.97 15319.72 13057.51
 (between_SS / total_SS =  91.5 %)
```

7

```
Console   Terminal   Background Jobs
> 8433 -y#
    Gender      Age   Height   Weight family_history_with_overweight      FAVC      FCVC      NCP
1 0.5457292 26.61366 1.701517  81.94530                      0.9081403 0.8948247 2.277071 2.506871
2 0.6177474 24.68997 1.745496 114.70637                      0.9965870 0.9829352 2.427736 2.861260
3 0.7157904 25.82921 1.791526 101.47240                      0.9846211 0.9799331 2.509572 2.416171
4 1.0000000 32.97757 1.802791 120.8438/                      1.0000000 0.9581333 2.553231 2.971673
5 0.2239024 20.20548 1.642104  49.41045                      0.4036145 0.7861446 2.376711 2.775990
6 0.0074268? 20.81589 1.751387 137.51050                      1.0000000 1.0000000 3.000000 1.000000
7 0.4822402? 21.75721 1.662681  65.13138                      0.6452514 0.7318436 2.345555 2.620479
      CAEC     SMOKE     CH2O      SCC      FAF      TUE     CALC     MTRANS Nobeyesdad  cluster
1 1.101475 0.01601441 1.989525 0.030005819 1.0035494 0.5788804 0.6928264 3.503441  2.908333 2.000000
2 1.031445 0.02021844 2.305511 0.000000000 0.9243440 0.7720700 0.8421405 3.789398  4.6182591 2.662207
3 1.003453 0.01541667 1.933594 0.000000000 0.8605816 0.5809106 0.9583333 3.917824  5.2118056 1.000000
5 1.500000 0.03506024 1.790860 0.108433735 1.1320485 0.8544768 0.6415663 3.804217  0.3222892 1.000000
6 1.007463 0.00000000 2.283175 0.000000000 0.0000000 0.0000000 0.0000000 3.000000  0.0000000 4.000000
7 1.142458 0.02793296 2.007240 0.111731844 1.1255260 0.6402621 0.6368715 3.458101  1.4497207 1.195531

Clustering vector:
  [1] 7 7 1 1 5 5 5 5 7 7 1 1 3 3 7 7 1 1 1 7 1 7 5 5 5 7 1 7 1 1 7 7 7 7 5 5 7 1 1 7 7 7 7 7 7 7 7 7
 [39] 5 5 5 5 7 7 7 5 5 7 7 1 3 2 7 1 5 5 7 7 1 1 1 7 1 1 1 7 1 3 7 5 1 1 1 7 1 1 1 7 1 1 1 7 1 1 1 7
[103] 5 7 1 1 1 5 5 7 1 1 1 5 5 7 1 1 1 5 5 7 1 1 1 5 5 7 1 1 1 5 5 7 1 1 1 5 5 7 1 1 1 5 5 7 1 1 1 7
[154] 1 1 7 5 7 7 1 7 1 7 1 1 4 7 1 1 1 1 7 7 7 5 1 7 7 1 3 7 5 1 1 7 1 1 7 5 7 1 1 3 7 5 1 1 7 1 1 7
[205] 7 1 1 1 5 7 1 1 1 5 7 1 1 1 5 7 1 1 1 5 7 1 1 1 5 7 1 1 1 5 7 1 1 1 5 7 1 1 1 5 7 1 1 1 5 5 5
[256] 1 1 2 1 1 1 7 5 7 5 5 1 7 3 5 7 7 5 5 7 5 7 5 7 3 5 7 5 1 7 7 7 2 7 5 1 1 7 5 5 5 3 3 7 5 5 5
[307] 7 1 1 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5
[358] 5 4 5 1 1 1 7 1 7 7 7 7 5 7 1 7 7 7 1 5 1 7 5 5 1 2 7 5 5 7 7 7 7 7 1 3 3 1 7 7 7
[409] 1 1 5 7 7 1 1 1 5 7 7 1 1 1 5 7 7 1 1 1 5 7 7 1 1 1 5 7 7 1 1 1 5 7 7 1 1 1 5 7 7 1 1 1 5 7 7 7
[460] 1 5 3 1 7 1 5 1 1 5 5 5 7 7 7 7 5 5 5 7 7 7 1 3 3 5 1 7 1 7 7 7 5 5 3 6 2 6 6 6 2 6 6 5 5 5
[511] 7 1 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 7 7
[562] 5 5 5 7 7 5 5 5 5 5 5 5 5 5 7 7 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[613] 5 5 5 5 7 7 1 1 1 1 5 7 5 1 7 7 1 1 1 1 5 7 5 1 7 7 1 1 1 1 5 7 5 1 7 7 1 1 1 1 5 7 5 1 7 7 1 1
[664] 5 5 5 5 7 7 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 7 7 5 5 7 7 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[715] 7 1 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 5 5 7 7 7 7
[766] 7 1 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1 1 7 1
[817] 1 1 1 1 7 1 1 1 7 1 1 1 7 1 1 1 7 1 1 1 7 1 1 1 7 1 1 1 7 1 1 1 7 1 1 1 7 1 1 1 7 1 1 1 7 1 7 7
[868] 7 7 7 1 1 1 7 7 7 7 7 7 7 7 1 1 1 1 1 1 1 1 7 5 1 1 7 7 1 1 1 1 1 1 1 1 1 1 1 7 7 7 7 7 7 7 7
[919] 1 1 7 7 7 1 1 1 7 7 7 1 1 1 7 7 7 1 1 1 7 7 7 1 1 1 5 5 7 7 1 1 5 5 7 7 1 1 5 5 7 7 1 1 7 7 7 1
[970] 1 1 1 1 7 1 7 1 1 1 1 1 1 1 7 1 7 1 1 7 1 7 1 1 1 1 1
[ reached getOption("max.print") -- omitted 1111 entries ]

Within cluster sum of squares by cluster:
[1] 4810.132 14808.640  1335.547 13435.978 12269.240  7686.491
[7] 10815.653 14643.281  5090.199
 (between_SS / total_SS =  94.5 %)
```

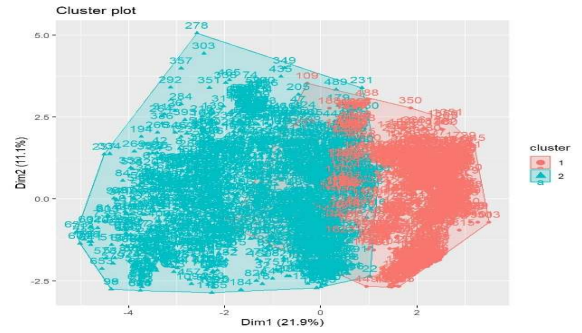
8

```
[769] 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784
  7   7   7   7   3   7   7   7   3   7   7   7   3   6   5   7
[785] 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800
  6   6   6   5   3   5   7   7   7   6   6   3   5   3   3   3
[801] 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816
  7   7   6   3   3   7   5   3   7   7   7   3   5   3   3   7
[817] 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832
  7   7   3   7   5   7   7   7   5   5   5   7   3   7   7   7
[833] 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848
  7   7   7   7   3   3   3   3   3   3   5   7   3   7   7   7
[849] 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864
  5   3   7   5   3   5   7   7   7   5   5   7   7   7   7
[865] 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880
  7   6   6   6   6   6   6   7   7   7   7   7   7   5   7   7   6
[881] 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896
  6   6   6   5   5   3   3   3   3   3   3   7   6   3   3   7
[897] 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912
  7   3   7   7   7   3   5   5   5   3   7   3   7   7   5   7
[913] 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928
  7   7   7   5   5   6   3   3   7   7   7   7   7   3   3   3
[929] 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944
  7   3   7   7   7   3   5   3   7   3   7   7   7   7   7   3
[945] 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960
  3   7   1   6   5   7   7   7   1   6   6   5   3   5   7   7
[961] 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976
  7   6   2   2   3   4   4   2   4   4   2   7   3   7   7   7
[977] 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992
  5   3   3   3   3   7   2   5   3   3   3   7   5   5   6   6   3
[993] 993 994 995 996 997 998 999 1000
  7   7   3   3   3   3   3   3
 [ reached getOption("max.print") -- omitted 1111 entries ]

Within cluster sum of squares by cluster:
[1] 8433.560 14102.869 11071.426 13435.978 11648.144  8834.667  7208.983
[8] 14808.640
 (between_SS / total_SS =  94.2 %)
```

9

```
[769] 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784
  5   5   1   1   1   1   1   1   1   5   5   5   1   1   5
[785] 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800
  1   1   1   1   1   5   5   5   5   5   5   5   5   5   5   5
[801] 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816
  5   5   5   5   5   1   5   5   5   1   1   5   5   5   1   1
[817] 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832
  1   1   1   1   1   5   5   5   5   5   5   5   5   5   5   5
[833] 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
[849] 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
[865] 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
[881] 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
[897] 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
[913] 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
[929] 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
[945] 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
[961] 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
[977] 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
[993] 993 994 995 996 997 998 999 1000
  5   5   5   5   5   5   5   5
 [ reached getOption("max.print") -- omitted 1111 entries ]

Within cluster sum of squares by cluster:
[1] 10815.653 14643.281
```

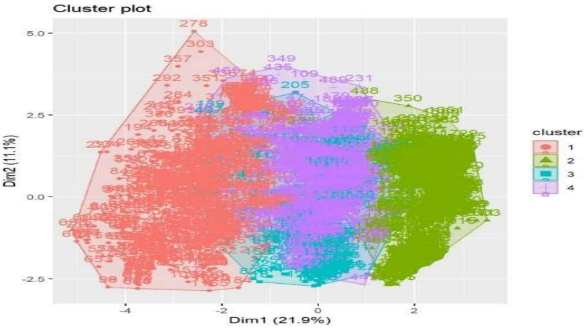
2


Cluster plot
Dim1 (21.9%)
Dim2 (11.1%)
cluster
1
2

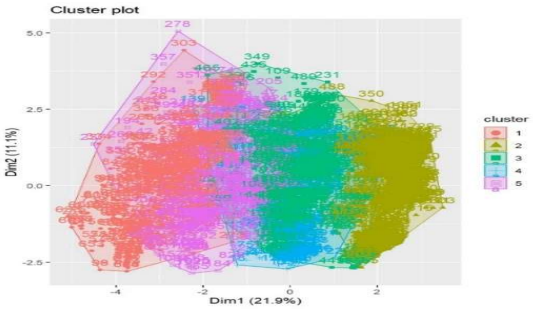
3

Cluster plot
Dim1 (21.9%)
Dim2 (11.1%)
cluster
1
2
3

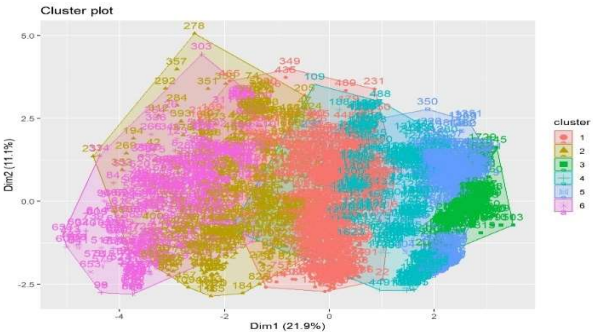
4


Cluster plot
Dim1 (21.9%)
Dim2 (11.1%)
cluster
1
2
3
4

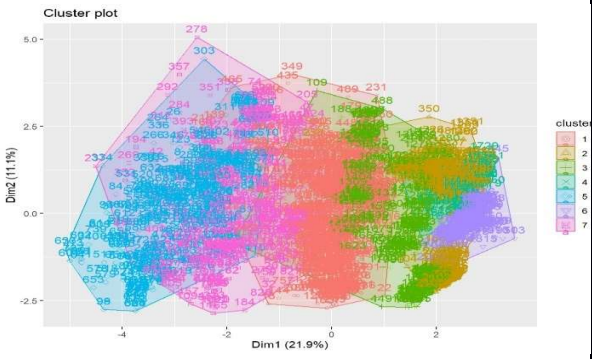
5


Cluster plot
Dim1 (21.9%)
Dim2 (11.1%)
cluster
1
2
3
4
5

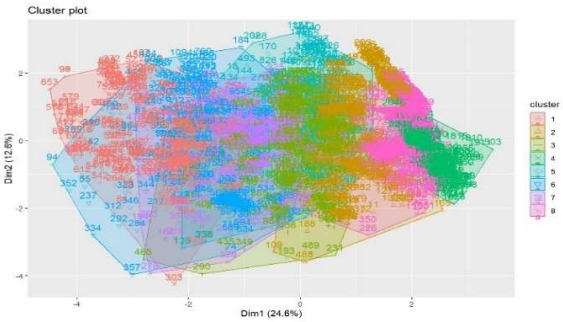
6


Cluster plot
Dim1 (21.9%)
Dim2 (11.1%)
cluster
1
2
3
4
5
6

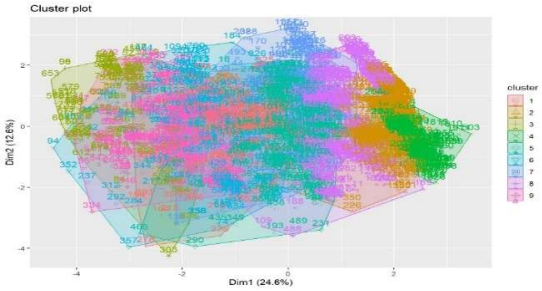
7


Cluster plot
Dim1 (21.9%)
Dim2 (11.1%)
1
2
3
4
5
6
7

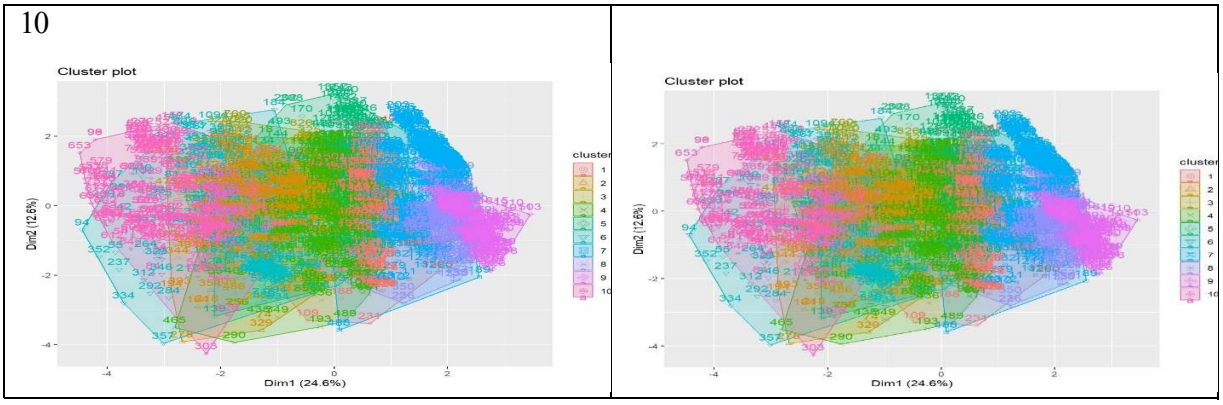
8

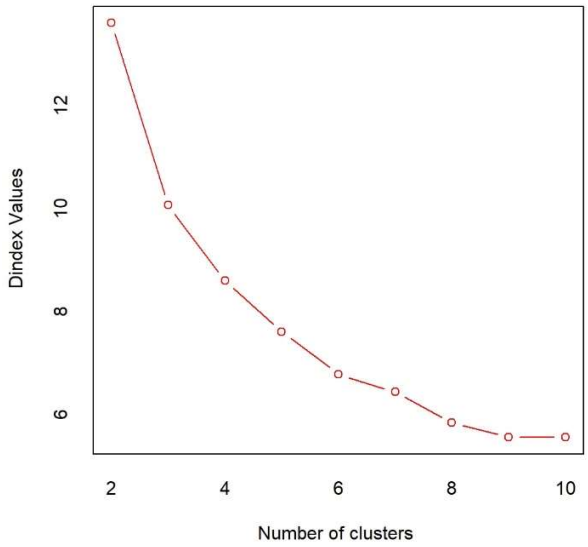

Cluster plot
Dim1 (24.6%)
Dim2 (12.6%)
cluster
1
2
3
4
5
6
7
8

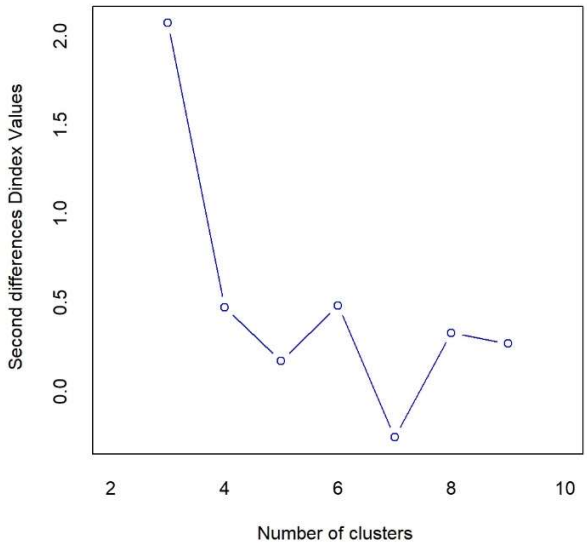
9


Cluster plot
Dim1 (24.6%)
Dim2 (12.6%)
cluster
1
2
3
4
5
6
7
8
9

10







Based on the plots above, we can conclude that 7, 8, or 9 are an optimal number of clusters.

3b) Build a table with your results succinctly displayed.

To perform K-means clustering on our data, we use kmeans(). We specify centers with different values to specify the number of clusters we expect. The resulting object contains the **Clustering Vector**, **Cluster means** and **between_SS /total_SS (%)** value. The collected findings for different centers are tabularized below:

Centers (No. of clusters to expect)	**between_SS /total_SS (%)**
2	67.4 %
3	81.7%
4	84.1%
5	86.8%
6	91.5%
7	91.9%
8	94.2 %
9	94.5%
10	95.1%

4. PREDICTION (USING TRAINING AND TEST SETS)

```r
# 4 PREDICTION 70-30
subset_data_70_30.train.k4 = kmeans(subset_data_70_30.train,centers = 8)
subset_data_70_30.train.k4

subset_data_70_30.test.k4 = knn(subset_data_70_30.train,subset_data_70_30.test,subset_data_70_30.train.k4$cluster, k=8)
subset_data_70_30.test.k4
length(subset_data_70_30.test.k4)

subset_data_70_30.test.kmeans.k4 = kmeans(subset_data_70_30.test,centers = 8)
subset_data_70_30.test.kmeans.k4
subset_data_70_30.test.k4.labels = subset_data_70_30.test.kmeans.k4$cluster
subset_data_70_30.test.k4.labels
length(subset_data_70_30.test.k4.labels)
#

# 4 PREDICTION 60-40
subset_data_60_40.train.k4 = kmeans(subset_data_60_40.train,centers = 8)
subset_data_60_40.train.k4

subset_data_60_40.test.k4 = knn(subset_data_60_40.train,subset_data_60_40.test,subset_data_60_40.train.k4$cluster, k=8)
subset_data_60_40.test.k4
length(subset_data_60_40.test.k4)

subset_data_60_40.test.kmeans.k4 = kmeans(subset_data_60_40.test,centers = 8)
subset_data_60_40.test.kmeans.k4
subset_data_60_40.test.k4.labels = subset_data_60_40.test.kmeans.k4$cluster
subset_data_60_40.test.k4.labels
length(subset_data_60_40.test.k4.labels)
#

# 4 PREDICTION 50-50
subset_data_50_50.train.k4 = kmeans(subset_data_50_50.train,centers = 8)
subset_data_50_50.train.k4

subset_data_50_50.test.k4 = knn(subset_data_50_50.train,subset_data_50_50.test,subset_data_50_50.train.k4$cluster, k=8)
subset_data_50_50.test.k4
length(subset_data_50_50.test.k4)

subset_data_50_50.test.kmeans.k4 = kmeans(subset_data_50_50.test,centers = 8)
subset_data_50_50.test.kmeans.k4
subset_data_50_50.test.k4.labels = subset_data_50_50.test.kmeans.k4$cluster
subset_data_50_50.test.k4.labels
length(subset_data_50_50.test.k4.labels)
#
```

4a) Use the glm(…) method to get linear fits for the data.
Try clusters of 5,7,9,11
Remember to generate labels using kmeans.
Do a cross-tabulation of the results to see how good you did by comparing kmeans vs prediction results.

Linear Modeling:
Linear modeling seeks to discover the optimal line that fits the given data. This methodology operates under the assumption that the problem can be analyzed feasibly, given the presumption of a linear relationship. Regression analysis is employed to discern potential dependencies.

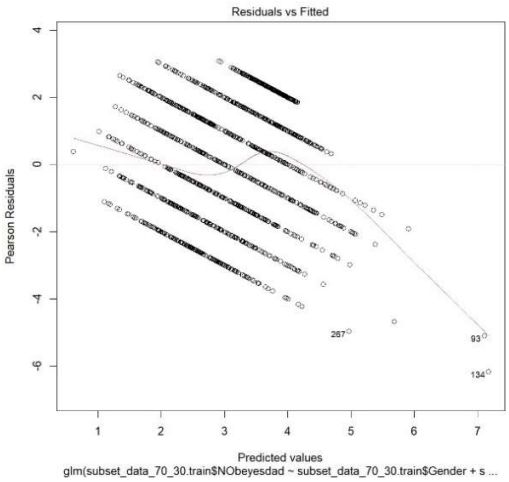
To execute this, we will utilize the glm() function provided by the stats package, which is automatically loaded by RStudio. Within glm, we employ symbols to define the models. Generally, a model is expressed as response ~ terms, where terms delineate a linear predictor for the response, and the response is represented by a numeric vector.

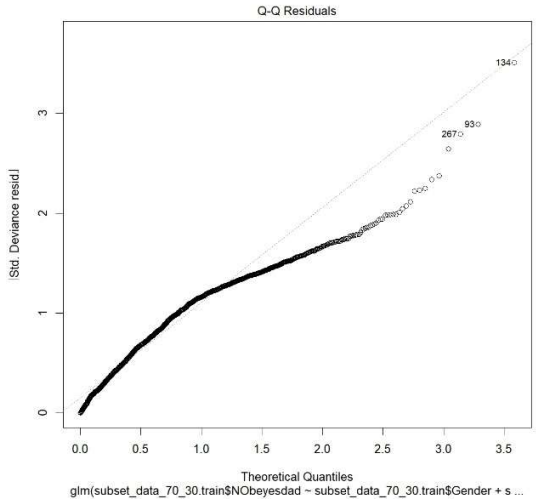
```r
# 4 PREDICTION 70-30
subset_data_70_30.train.k8 = kmeans(subset_data_70_30.train,centers = 8)
subset_data_70_30.train.k8

subset_data_70_30.test.k8 = knn(subset_data_70_30.train,subset_data_70_30.test,subset_data_70_30.train.k8$cluster, k=8)
subset_data_70_30.test.k8
length(subset_data_70_30.test.k8)

subset_data_70_30.test.kmeans.k8 = kmeans(subset_data_70_30.test,centers = 8)
subset_data_70_30.test.kmeans.k8
subset_data_70_30.test.k8.labels = subset_data_70_30.test.kmeans.k8$cluster
subset_data_70_30.test.k8.labels
length(subset_data_70_30.test.k8.labels)

#Linear Modelling
subset_data_70_30.train.glm = glm(formula = subset_data_70_30.trainNObeyesdad~subset_data_70_30.trainGender+subset_data_70_30.trainHeight+subset_data_70_30.trainAge+subset_data_70_30.trainFCVC+subset_data_70_30.trainCH2O+subset_data_70_30.trainCALC+subset_data_70_30.trainTU,family = gaussian,
summary(subset_data_70_30.train.glm)
plot(subset_data_70_30.train.glm)

subset_data_70_30.train.glm.anova = anova(subset_data_70_30.train.glm,test="Chisq")
subset_data_70_30.train.glm.anova

#predict
subset_data_70_30.test.pred = predict(subset_data_70_30.train.glm,newdata = subset_data_70_30.test)
subset_data_70_30.test.pred
summary(subset_data_70_30.test.pred)

#Confidence Intervals:
confint(subset_data_70_30.train.glm)


#

# 4 PREDICTION 60-40
subset_data_60_40.train.k8 = kmeans(subset_data_60_40.train,centers = 8)
subset_data_60_40.train.k8

subset_data_60_40.test.k8 = knn(subset_data_60_40.train,subset_data_60_40.test,subset_data_60_40.train.k8$cluster, k=8)
subset_data_60_40.test.k8
length(subset_data_60_40.test.k8)

subset_data_60_40.test.kmeans.k8 = kmeans(subset_data_60_40.test,centers = 8)
subset_data_60_40.test.kmeans.k8
subset_data_60_40.test.k8.labels = subset_data_60_40.test.kmeans.k8$cluster
subset_data_60_40.test.k8.labels
length(subset_data_60_40.test.k8.labels)

#Linear Modelling
subset_data_60_40.train.glm = glm(formula = subset_data_60_40.trainNObeyesdad~subset_data_60_40.trainGender+subset_data_60_40.trainHeight+subset_data_60_40.trainAge+subset_data_60_40.trainFCVC+subset_data_60_40.trainCH2O+subset_data_60_40.trainCALC+subset_data_60_40.trainTUE,family = gaussian
summary(subset_data_60_40.train.glm)
plot(subset_data_60_40.train.glm)
subset_data_60_40.train.glm.anova = anova(subset_data_60_40.train.glm,test="Chisq")
subset_data_60_40.train.glm.anova

#predict
subset_data_60_40.test.pred = predict(subset_data_60_40.train.glm,newdata = subset_data_60_40.test)
subset_data_60_40.test.pred
summary(subset_data_60_40.test.pred)

#Confidence Intervals:
confint(subset_data_60_40.train.glm)
```

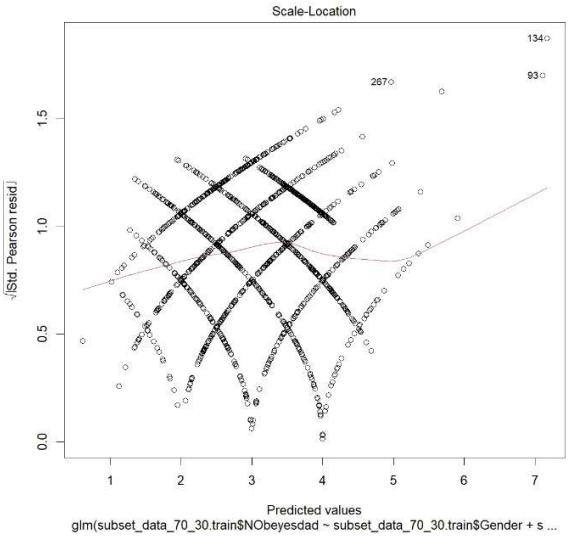
```r

# 4 PREDICTION 50-50
subset_data_50_50.train.k8 = kmeans(subset_data_50_50.train,centers = 8)
subset_data_50_50.train.k8

subset_data_50_50.test.k8 = knn(subset_data_50_50.train,subset_data_50_50.test,subset_data_50_50.train.k8$cluster, k=8)
subset_data_50_50.test.k8
length(subset_data_50_50.test.k8)

subset_data_50_50.test.kmeans.k8 = kmeans(subset_data_50_50.test,centers = 8)
subset_data_50_50.test.kmeans.k8
subset_data_50_50.test.k8.labels = subset_data_50_50.test.kmeans.k8$cluster
subset_data_50_50.test.k8.labels
length(subset_data_50_50.test.k8.labels)

#Linear Modelling
subset_data_50_50.train.glm = glm(formula = subset_data_50_50.trainNObeyesdad~subset_data_50_50.trainGender+subset_data_50_50.trainHeight+subset_data_50_50.trainAge+subset_data_50_50.trainFCVC+subset_data_50_50.trainCH2O+subset_data_50_50.trainCALC+subset_data_50_50.trainTUE,family = gaussian
summary(subset_data_50_50.train.glm)
plot(subset_data_50_50.train.glm)
subset_data_50_50.train.glm.anova = anova(subset_data_50_50.train.glm,test="Chisq")
subset_data_50_50.train.glm.anova

#Predict
subset_data_50_50.test.pred = predict(subset_data_50_50.train.glm,newdata = subset_data_50_50.test)
subset_data_50_50.test.pred
summary(subset_data_50_50.test.pred)

#Confidence Intervals:
confint(subset_data_50_50.train.glm)
#
```

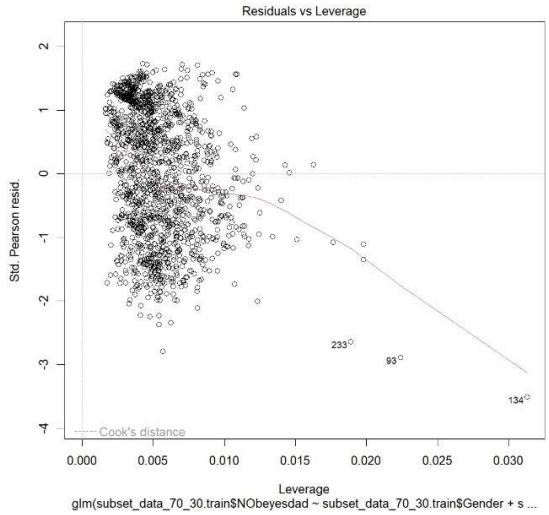
70-30 SPLIT

```
Console   Terminal ×   Background Jobs ×
R 4.3.3 · ~/
> summary(subset_data_70_30.train.glm)

Call:
glm(formula = subset_data_70_30.train$NObeyesdad ~ subset_data_70_30.train$Gender +
    subset_data_70_30.train$Height + subset_data_70_30.train$Age +
    subset_data_70_30.train$FCVC + subset_data_70_30.train$CH2O +
    subset_data_70_30.train$CALC + subset_data_70_30.train$TU,
    family = gaussian, data = subset_data_70_30.train)

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -8.125301   1.082445  -7.506 1.05e-13 ***
subset_data_70_30.train$Gender  -0.444261   0.126715  -3.506 0.000469 ***
subset_data_70_30.train$Height   4.041374   0.661917   6.106 1.31e-09 ***
subset_data_70_30.train$Age      0.088438   0.007649  11.562  < 2e-16 ***
subset_data_70_30.train$FCVC     0.691914   0.093338   7.413 2.08e-13 ***
subset_data_70_30.train$CH2O     0.264389   0.078165   3.382 0.000737 ***
subset_data_70_30.train$CALC     0.374003   0.091669   4.080 4.75e-05 ***
subset_data_70_30.train$TU      -0.119403   0.079789  -1.496 0.134740
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 3.192094)

    Null deviance: 5739.6  on 1476  degrees of freedom
Residual deviance: 4689.2  on 1469  degrees of freedom
AIC: 5915.8

Number of Fisher Scoring iterations: 2

> subset_data_70_30.train.glm.anova = anova(subset_data_70_30.train.glm,test="chisq")
> subset_data_70_30.train.glm.anova
Analysis of Deviance Table

Model: gaussian, link: identity

Response: subset_data_70_30.train$NObeyesdad

Terms added sequentially (first to last)


                                 Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                              1476      5739.6
subset_data_70_30.train$Gender  1      1.86      1475      5737.7 0.4449922
subset_data_70_30.train$Height  1    199.08      1474      5538.6 2.851e-15 ***
subset_data_70_30.train$Age     1    543.17      1473      4995.5 < 2.2e-16 ***
subset_data_70_30.train$FCVC    1    204.49      1472      4791.0 1.206e-15 ***
subset_data_70_30.train$CH2O    1     41.41      1471      4749.6 0.0003158 ***
subset_data_70_30.train$CALC    1     53.24      1470      4696.3 4.425e-05 ***
subset_data_70_30.train$TU      1      7.15      1469      4689.2 0.1345255
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```



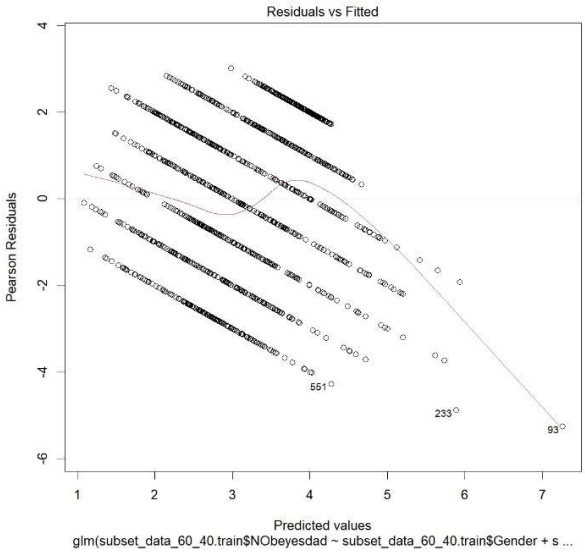


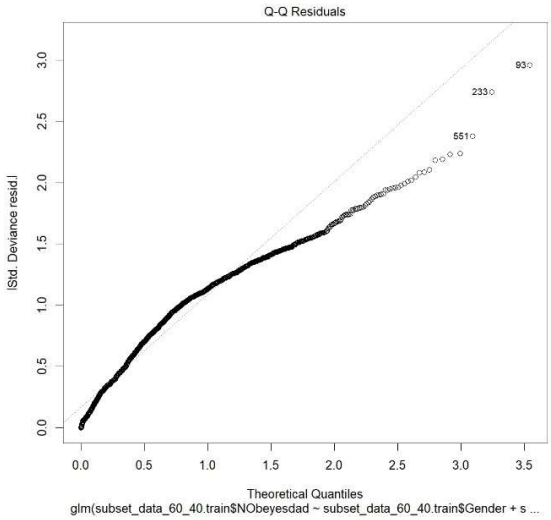


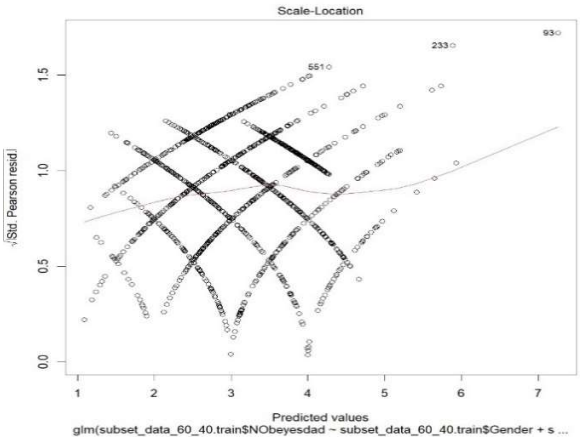


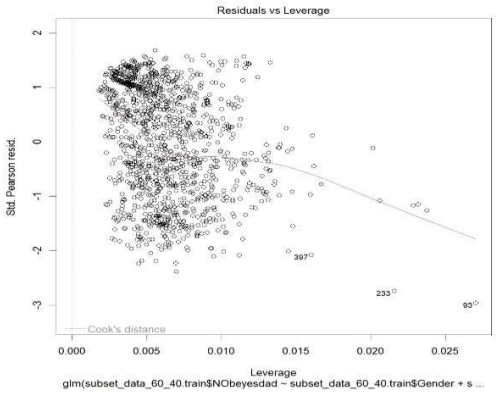
60-40 SPLIT

```
> summary(subset_data_60_40.train.glm)

Call:
glm(formula = subset_data_60_40.train$NObeyesdad ~ subset_data_60_40.train$Gender +
    subset_data_60_40.train$Height + subset_data_60_40.train$Age +
    subset_data_60_40.train$FCVC + subset_data_60_40.train$CH2O +
    subset_data_60_40.train$CALC + subset_data_60_40.train$TUE,
    family = gaussian, data = subset_data_60_40.train)

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    -6.376361   1.183478  -5.388 8.51e-08 ***
subset_data_60_40.train$Gender -0.356134   0.136178  -2.615 0.009024 **
subset_data_60_40.train$Height  2.876897   0.720142   3.995 6.85e-05 ***
subset_data_60_40.train$Age     0.089361   0.008411  10.624  < 2e-16 ***
subset_data_60_40.train$FCVC    0.711646   0.099870   7.126 1.24e-12 ***
subset_data_60_40.train$CH2O    0.330670   0.085056   3.888 0.000106 ***
subset_data_60_40.train$CALC    0.414240   0.099906   4.146 3.61e-05 ***
subset_data_60_40.train$TUE    -0.058043   0.087421  -0.664 0.506845
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 3.241665)

    Null deviance: 4955.1  on 1265  degrees of freedom
Residual deviance: 4078.0  on 1258  degrees of freedom
AIC: 5091.7

Number of Fisher Scoring iterations: 2

> subset_data_60_40.train.glm.anova
Analysis of Deviance Table

Model: gaussian, link: identity

Response: subset_data_60_40.train$NObeyesdad

Terms added sequentially (first to last)

                                Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                             1     12.00      1264      4943.1   0.05431 .
subset_data_60_40.train$Gender   1    113.26      1263      4829.9 3.402e-09 ***
subset_data_60_40.train$Age      1    447.47      1262      4382.4 < 2.2e-16 ***
subset_data_60_40.train$FCVC     1    192.21      1261      4190.2 1.357e-14 ***
subset_data_60_40.train$CH2O     1     54.26      1260      4135.9 4.288e-05 ***
subset_data_60_40.train$CALC     1     56.46      1259      4079.4 3.002e-05 ***
subset_data_60_40.train$TUE      1      1.43      1258      4078.0   0.50672
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```



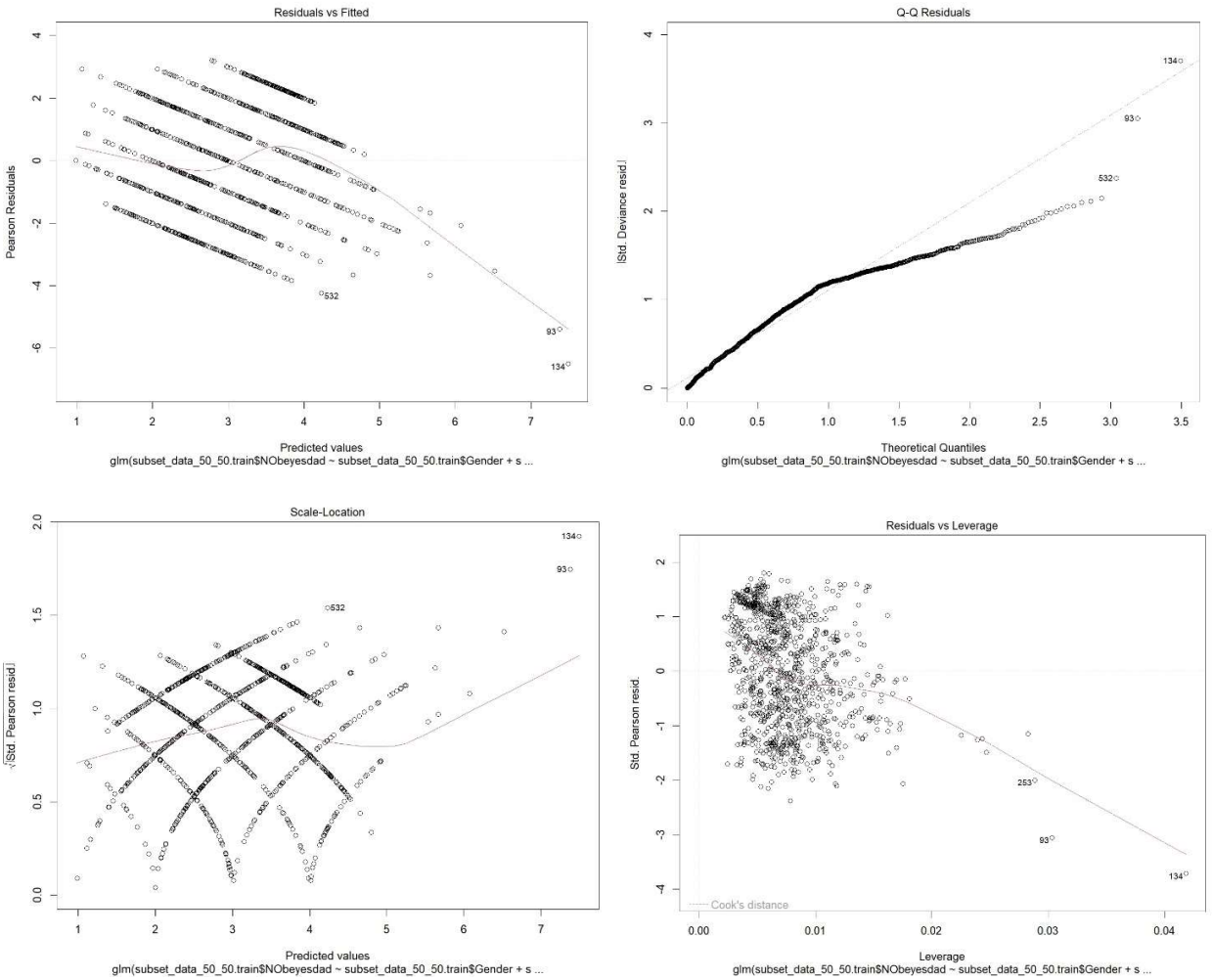






50-50 SPLIT

```
> summary(subset_data_50_50.train.glm)
Call:
glm(formula = subset_data_50_50.train$Nobeyesdad ~ subset_data_50_50.train$Gender +
    subset_data_50_50.train$Height + subset_data_50_50.train$Age +
    subset_data_50_50.train$FCVC + subset_data_50_50.train$CH2O +
    subset_data_50_50.train$CALC + subset_data_50_50.train$TUE.
    family = gaussian, data = subset_data_50_50.train)
Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                        -18.30275    1.19091  -15.437  < 2e-16 ***
subset_data_50_50.train$Gender     -0.39663    0.15098   -2.627  0.008737 **
subset_data_50_50.train$Height     -3.31687    0.75407   -4.399  9.44e-07 ***
subset_data_50_50.train$Age        0.09787    0.00848   11.541  < 2e-16 ***
subset_data_50_50.train$FCVC       0.67413    0.10989    6.135  1.2e-09 ***
subset_data_50_50.train$CH2O       0.35140    0.09203    3.819  0.000152 ***
subset_data_50_50.train$CALC       0.36761    0.10813    3.399  0.000724 ***
subset_data_50_50.train$TUE        -0.03778    0.09550   -0.396  0.692482
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for gaussian family taken to be 3.203062)
    Null deviance: 4173.1  on 1054  degrees of freedom
Residual deviance: 3353.6  on 1047  degrees of freedom
AIC: 4232.1
Number of Fisher Scoring iterations: 2
> subset_data_50_50.train.glm.anova
Analysis of Deviance Table
Model: gaussian, link: identity
Response: subset_data_50_50.train$Nobeyesdad
Terms added sequentially (first to last)
                                    Df Deviance Resid. Df Resid. Dev   Pr(>Chi)
NULL                                             1054     4173.0
subset_data_50_50.train$Gender      1     0.05      1053     4173.0 0.8926456
subset_data_50_50.train$Height      1   138.99      1052     4034.0 4.478e-11 ***
subset_data_50_50.train$Age         1   451.11      1051     3583.0 < 2.2e-16 ***
subset_data_50_50.train$FCVC        1   137.22      1050     3445.7 5.942e-11 ***
subset_data_50_50.train$CH2O        1    41.52      1049     3404.2 0.00031** ***
subset_data_50_50.train$CALC        1    37.08      1048     3367.1 0.00066** ***
subset_data_50_50.train$TUE         1     0.50      1047     3353.6 0.692401
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



Graph reveal informative characteristics of the data. The Q-Q plot suggests a largely normal distribution with potential outliers at the upper tail. The Spread Location Plot indicates homoscedasticity, signifying constant variance across the data. An examination of the Residuals vs. Leverage Plot, which identifies influential points, confirms the absence of observations exceeding Cook's distance threshold, implying minimal impact on the regression model.

While outliers may still be present, their influence appears negligible. Therefore, predictions can be made with a degree of confidence, acknowledging the presence of these outliers. The application of the model to the training set will yield a vector of predicted values for the target variable, 'Rings,' facilitating further analysis and informed decision-making.

PREDICT

70-30 SPLIT

```
> #Predict
> subset_data_70_30.test.pred = predict(subset_data_70_30.train.glm,newdata = subset_data_70_30.test)
Warning message:
'newdata' had 634 rows but variables found have 1477 rows
> subset_data_70_30.test.pred
       1        2        3        4        5        6        7        8        9       10       11       12       13       14       15       16       17       18       19       20
3.997721 3.465715 2.522368 3.368239 2.629091 3.431465 3.746909 4.149773 4.665125 1.654407 2.385651 2.567416 2.629742 3.872968 4.385220 4.209353 2.469765 2.326765 3.247257 2.622184
      21       22       23       24       25       26       27       28       29       30       31       32       33       34       35       36       37       38       39       40
3.491726 1.525090 2.542168 3.457853 5.907898 1.791325 3.182630 2.385651 2.315156 4.051855 2.810510 1.479622 4.347680 2.385651 1.191991 4.316085 2.700162 2.297541 3.905402 1.945293
      41       42       43       44       45       46       47       48       49       50       51       52       53       54       55       56       57       58       59       60
2.372395 3.266251 4.454009 4.123004 3.861602 3.120582 1.657634 3.637230 3.751666 4.648273 2.494434 2.242213 3.061642 1.301770 2.190374 2.785006 2.787828 2.871172 4.153121 3.703192
      61       62       63       64       65       66       67       68       69       70       71       72       73       74       75       76       77       78       79       80
3.813841 3.194193 3.362009 1.559150 2.422232 2.110947 3.079244 4.996336 3.900176 1.527195 5.483484 4.793681 1.994959 3.647443 3.265961 5.144133 3.230267 2.531804 2.446783 3.858631
      81       82       83       84       85       86       87       88       89       90       91       92       93       94       95       96       97       98       99      100
3.619396 3.513599 2.351640 2.163482 3.865224 3.451616 3.226098 3.611415 2.665252 2.931375 3.251398 3.654992 5.063890 3.173140 2.768932 1.727732 3.050859 3.700072 3.789250 3.809257
     101      102      103      104      105      106      107      108      109      110      111      112      113      114      115      116      117      118      119      120
3.651525 3.637814 2.525108 2.933525 4.398880 3.893642 3.255623 3.276234 2.601790 3.013209 3.962948 2.165329 2.670569 3.608573 3.904748 2.702032 2.259358 2.725356 4.464526 2.234603
```

```
> summary(subset_data_70_30.test.pred)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.6099  2.4399  3.1177  3.0840  3.7360  7.1708
```

60-40 SPLIT

```
> #Predict
> subset_data_60_40.test.pred = predict(subset_data_60_40.train.glm,newdata = subset_data_60_40.test)
Warning message:
'newdata' had 845 rows but variables found have 1266 rows
> subset_data_60_40.test.pred
       1        2        3        4        5        6        7        8        9       10       11       12       13       14       15       16       17       18       19       20
3.318304 3.287421 3.730220 3.874467 1.873330 2.843253 3.578063 4.097190 3.483520 2.644266 4.152376 4.025842 2.380174 1.086284 4.338500 3.990743 3.910100 3.326159 2.740118 3.550536
      21       22       23       24       25       26       27       28       29       30       31       32       33       34       35       36       37       38       39       40
4.075870 3.862543 3.668665 2.637214 2.153818 2.634205 4.286280 3.083850 2.882104 3.312104 3.398501 2.190868 2.520245 2.397694 3.657897 3.543466 2.563694 2.636694 1.819077 2.857372
      41       42       43       44       45       46       47       48       49       50       51       52       53       54       55       56       57       58       59       60
3.118781 3.971707 3.773379 3.844201 2.306862 2.688605 1.724259 4.711587 3.029760 3.019768 3.531948 3.116639 5.158220 3.580405 3.570839 3.718888 3.012617 1.786312 1.481759 2.768632
      61       62       63       64       65       66       67       68       69       70       71       72       73       74       75       76       77       78       79       80
3.581355 1.530588 2.471589 3.487489 1.987911 1.811200 3.651255 2.923979 3.994120 2.610964 4.086571 4.032068 2.040851 4.901621 3.507481 3.994550 2.596622 2.887501 3.546982 1.782559
      81       82       83       84       85       86       87       88       89       90       91       92       93       94       95       96       97       98       99      100
3.919898 1.805580 2.648810 2.741667 4.401108 5.882272 4.368225 1.629571 3.226954 2.705847 3.115757 3.005090 2.118471 2.919900 2.508203 3.854699 2.400796 3.748808 2.443737 3.371914
     101      102      103      104      105      106      107      108      109      110      111      112      113      114      115      116      117      118      119      120
2.390710 2.855154 2.252238 2.384992 3.364336 1.714360 2.575322 4.092138 2.417283 1.805049 2.578220 1.882874 3.924033 3.885397 1.927407 4.447556 3.407685 2.605739 3.532331 2.285227
     121      122      123      124      125      126      127      128      129      130      131      132      133      134      135      136      137      138      139      140
3.424948 4.714598 2.634205 2.391547 2.807121 1.524985 3.674789 1.243772 3.413549 2.523813 1.785888 3.849943 3.150922 2.730368 3.643629 4.970041 2.268657 2.683958 1.897620 2.643043
     141      142      143      144      145      146      147      148      149      150      151      152      153      154      155      156      157      158      159      160
```

```
> summary(subset_data_60_40.test.pred)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.086   2.566   3.193   3.173   3.796   7.259
>
```

50-50 SPLIT

```
> subset_data_50_50.test.pred = predict(subset_data_50_50.train.glm,newdata = subset_data_50_50.test)
Warning message:
'newdata' had 1056 rows but variables found have 1055 rows
> subset_data_50_50.test.pred
       1        2        3        4        5        6        7        8        9       10       11       12       13       14       15       16       17       18       19       20       21
2.9319169 3.2756825 3.1594708 3.2918681 2.4214705 2.0619776 3.6658288 2.1742709 2.0728378 2.5054079 3.5827417 4.1095095 3.4759127 2.5115095 3.3687928 2.3882550 2.4251687 2.7526745 2.6132409 2.8613300 3.8292121
      22       23       24       25       26       27       28       29       30       31       32       33       34       35       36       37       38       39       40       41       42
2.7306336 3.7492451 4.5772256 2.7539067 2.2748232 3.1431130 2.2797385 2.0608242 3.7442980 1.7387856 3.3868867 3.9034094 1.8610130 3.3437059 2.9292978 2.7539067 2.3095322 1.9438283 2.7450587 1.5535913 2.9141812
      43       44       45       46       47       48       49       50       51       52       53       54       55       56       57       58       59       60       61       62       63
5.0789053 2.7405355 1.9611233 3.1642260 4.7075743 3.0998308 4.2251014 3.3974475 3.8446958 3.3461834 2.2287458 3.7731548 4.8192205 1.9525610 3.9665417 2.6668088 3.3215834 1.7198332 1.9714037 2.4043814 2.1841060
      64       65       66       67       68       69       70       71       72       73       74       75       76       77       78       79       80       81       82       83       84
2.0650959 2.3264586 2.3807163 3.6070595 2.3112683 2.7625606 2.9493291 2.6132409 3.3317341 2.4685421 2.2531219 3.0617352 2.2123954 2.9769863 2.3237873 3.8562067 4.1071607 2.2785445 2.3628505 2.3356572 2.4514500
      85       86       87       88       89       90       91       92       93       94       95       96       97       98       99      100      101      102      103      104      105
3.3999212 3.4473260 3.2243391 3.1489475 3.9642355 1.6487865 2.8479433 3.0582979 2.9958711 2.3345773 1.8297130 2.4179436 1.4653141 3.3958180 1.9545478 4.4667085 3.5792005 2.5746806 3.0417435 3.0734993 4.0176818
     106      107      108      109      110      111      112      113      114      115      116      117      118      119      120      121      122      123      124      125      126
3.7791778 3.2404282 3.0324164 2.7105502 4.2613469 2.5935435 4.2729374 2.5889759 2.0844637 3.6625160 3.0837770 3.1154607 3.2293652 1.6982622 4.8781870 2.3617667 4.1075050 3.3481031 2.6276211 3.4915872 4.4058854
     127      128      129      130      131      132      133      134      135      136      137      138      139      140      141      142      143      144      145      146      147
2.9581184 3.0998806 2.5738328 3.3806986 3.1507878 3.4775521 3.2404846 1.4172064 2.3809550 2.1630441 1.9277745 2.6968334 3.4624349 1.9354931 4.3224113 1.8675883 3.5487014 2.1971873 2.4348961 2.7789120 1.8411277
     148      149      150      151      152      153      154      155      156      157      158      159      160      161      162      163      164      165      166      167      168
2.7068746 3.2272577 1.8479902 2.5310963 3.8337923 4.0887683 2.5451285 2.9917393 3.8207205 4.2105708 3.7749690 3.8224998 4.3169147 1.5788129 3.8223372 2.5652675 1.0678684 4.0059223 2.0459384 2.1772515 1.9901185
     169      170      171      172      173      174      175      176      177      178      179      180      181      182      183      184      185      186      187      188      189
3.6750037 3.2727826 2.6109394 1.7233389 3.7384316 3.3579023 1.2160868 2.6305863 2.6520282 2.2163557 2.6393050 3.5511334 2.6505700 2.4418674 2.4569471 4.0847721 2.8346562 3.4470522 1.6188628 2.3809550 3.1810615
     190      191      192      193      194      195      196      197      198      199      200      201      202      203      204      205      206      207      208      209      210
3.9852899 3.1853435 3.2137558 1.8678286 3.8369548 3.3914077 7.4929255 3.9952587 2.0639956 2.2674338 3.4565530 2.3456785 1.7618804 3.6330360 2.1935304 2.4440446 2.4488876 2.0664072 2.5390816 2.6501659 3.1385160
     211      212      213      214      215      216      217      218      219      220      221      222      223      224      225      226      227      228      229      230      231
```

```
> summary(subset_data_50_50.test.pred)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.9858  2.3899  3.0604  3.0682  3.7150  7.4929
>
```

CONFIDENCE INTERVALS

70-30 SPLIT

```
> #Confidence Intervals:
> confint(subset_data_70_30.train.glm)
Waiting for profiling to be done...
                                      2.5 %     97.5 %
(Intercept)                     -10.24685546 -6.0037471
subset_data_70_30.train$Gender   -0.69261873 -0.1959039
subset_data_70_30.train$Height    2.74404034  5.3387069
subset_data_70_30.train$Age       0.07344594  0.1034294
subset_data_70_30.train$FCVC      0.50897478  0.8748524
subset_data_70_30.train$CH2O      0.11118799  0.4175899
subset_data_70_30.train$CALC      0.19433560  0.5536704
subset_data_70_30.train$TU       -0.27578652  0.0369799
>
```

60-40 SPLIT

```
> #Confidence Intervals:
> confint(subset_data_60_40.train.glm)
Waiting for profiling to be done...
                                      2.5 %      97.5 %
(Intercept)                     -8.69593442 -4.05678691
subset_data_60_40.train$Gender  -0.62303790 -0.08923054
subset_data_60_40.train$Height   1.46544502  4.28834939
subset_data_60_40.train$Age      0.07287611  0.10584619
subset_data_60_40.train$FCVC     0.51590395  0.90738705
subset_data_60_40.train$CH2O     0.16396345  0.49737701
subset_data_60_40.train$CALC     0.21842855  0.61005210
subset_data_60_40.train$TUE     -0.22938543  0.11329916
>
```

50-50 SPLIT

```
> #Confidence Intervals:
> confint(subset_data_50_50.train.glm)
Waiting for profiling to be done...
                                      2.5 %     97.5 %
(Intercept)                     -10.83288916 -5.7726188
subset_data_50_50.train$Gender   -0.69253910 -0.1007245
subset_data_50_50.train$Height    2.36032521  5.4730093
subset_data_50_50.train$Age       0.08089531  0.1148438
subset_data_50_50.train$FCVC      0.45885229  0.8900088
subset_data_50_50.train$CH2O      0.14085355  0.5015535
subset_data_50_50.train$CALC      0.15508516  0.5801293
subset_data_50_50.train$TUE      -0.22494837  0.1493918
>
```

COMPARING ACTUAL VS. PREDICTION

70-30 SPLIT

```
> #Compare Actual Vs Prediction
> subset_data_70_30.test.pred.k8 = kmeans(subset_data_70_30.test.pred,centers = 8)
> subset_data_70_30.test.pred.k8
K-means clustering with 8 clusters of sizes 46, 88, 147, 37, 59, 104, 48, 105

Cluster means:
        [,1]
1  6.4592076
2  2.8816343
3  5.1808267
4 -0.1440198
5  0.4091717
6  2.1889937
7  1.2939569
8  3.7702083

Clustering vector:
   2    4   10   12   15   17   19   21   22   31   35   36   46   47
   6    6    7    6    5    2    2    2    8    6    7    2    2    7
  50   53   55   58   62   65   66   69   71   74   75   82   88   89
   7    5    5    6    5    7    6    3    2    4    2    2    6    7
  96   97   99  104  107  117  119  122  124  136  137  142  144  146
   7    7    5    7    7    2    7    6    6    6    7    4    6    6
 154  159  162  166  167  169  173  176  183  193  199  200  205  207
   2    2    2    3    7    6    7    6    5    6    5    7    7    8
 213  215  216  217  219  220  221  224  226  229  232  236  240  241
   2    7    6    7    7    6    6    6    3    2    6    6    7    7
 244  245  255  256  257  260  261  268  269  272  278  279  281  282
```

```
Within cluster sum of squares by cluster:
[1]  9.860007  3.336127 15.265696  2.421694  1.740814  3.713054
[7]  2.740511  7.837549
 (between_SS / total_SS =  98.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
```

```
> subset_data_70_30.test.ct.k8 = CrossTable(subset_data_70_30.test.pred.k8$cluster,subset_data_70_30.test.kmeans.k8$cluster,prop.chisq = TRUE)


   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|
```

Total Observations in Table: 634

subset_data_70_30.test.pred.k8$cluster	subset_data_70_30.test.kmeans.k8$cluster 1	2	3	4	5	6	7	8	Row Total
1	0 6.095 0.000 0.000 0.000	0 6.893 0.000 0.000 0.000	5 1.530 0.109 0.042 0.008	0 4.281 0.000 0.000 0.000	0 1.741 0.000 0.000 0.000	0 9.069 0.000 0.000 0.000	41 404.161 0.891 0.854 0.065	0 5.804 0.000 0.000 0.000	46 0.073
2	9 0.607 0.102 0.107 0.014	0 13.186 0.000 0.000 0.000	0 16.517 0.000 0.000 0.000	24 30.525 0.273 0.407 0.038	8 6.543 0.091 0.333 0.013	36 20.047 0.409 0.288 0.057	0 6.662 0.000 0.000 0.000	11 0.001 0.125 0.138 0.017	88 0.139
3	0 19.476 0.000 0.000 0.000	0 22.027 0.000 0.000 0.000	109 240.195 0.741 0.916 0.172	0 13.680 0.000 0.000 0.000	0 5.565 0.000 0.000 0.000	0 28.983 0.000 0.000 0.000	7 1.532 0.048 0.146 0.011	31 8.358 0.211 0.388 0.049	147 0.232
4	10 5.301 0.270 0.119 0.016	27 83.034 0.730 0.284 0.043	0 6.945 0.000 0.000 0.000	0 3.443 0.000 0.000 0.000	0 1.401 0.000 0.000 0.000	0 7.295 0.000 0.000 0.000	0 2.801 0.000 0.000 0.000	0 4.669 0.000 0.000 0.000	37 0.058
5	11 1.296 0.186 0.131 0.017	47 164.708 0.797 0.495 0.074	0 11.074 0.000 0.000 0.000	0 5.491 0.000 0.000 0.000	0 2.233 0.000 0.000 0.000	1 9.718 0.017 0.008 0.002	0 4.467 0.000 0.000 0.000	0 7.445 0.000 0.000 0.000	59 0.093
6	31 21.522 0.298 0.369 0.049	3 10.161 0.029 0.032 0.005	0 19.521 0.000 0.000 0.000	4 3.331 0.038 0.068 0.006	11 12.672 0.106 0.458 0.017	54 54.716 0.519 0.432 0.085	0 7.874 0.000 0.000 0.000	1 11.199 0.010 0.012 0.002	104 0.164
7	23 43.541 0.479 0.274 0.036	18 16.240 0.375 0.189 0.028	0 9.009 0.000 0.000 0.000	0 4.467 0.000 0.000 0.000	5 5.576 0.104 0.208 0.008	2 5.886 0.042 0.016 0.003	0 3.634 0.000 0.000 0.000	0 6.057 0.000 0.000 0.000	48 0.076
8	0 13.912 0.000 0.000 0.000	0 15.733 0.000 0.000 0.000	5 10.977 0.048 0.042 0.008	31 46.121 0.295 0.525 0.049	0 3.975 0.000 0.000 0.000	32 6.166 0.305 0.256 0.050	0 7.950 0.000 0.000 0.000	37 42.576 0.352 0.463 0.058	105 0.166
Column Total	84 0.132	95 0.150	119 0.188	59 0.093	24 0.038	125 0.197	48 0.076	80 0.126	634

Cluster	subset_data_70_30.test.kmeans.k8	subset_data_70_30.test.pred.k8
1	**1**	**1**
2	**95**	**88**
3	**119**	**147**
4	**59**	**37**
5	**24**	**59**
6	**125**	**104**
7	**48**	**48**
8	**80**	**105**

60-40 SPLIT

```
> #Compare Actual Vs Prediction
> subset_data_60_40.test.pred.k8 = kmeans(subset_data_60_40.test.pred,centers = 8)
> subset_data_60_40.test.pred.k8
K-means clustering with 8 clusters of sizes 194, 103, 36, 169, 154, 238, 185, 187

Cluster means:
      [,1]
1 3.313539
2 1.707513
3 4.873921
4 2.955629
5 4.161372
6 3.746280
7 2.227848
8 2.608125

Clustering vector:
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26
   6   8   4   6   1   3   2   5   8   7   1   2   8   5   6   6   6   7   8   7   5   5   6   4   3   4
  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52
   4   8   6   5   3   4   8   2   6   8   7   1   6   6   5   2   8   5   5   8   6   2   2   7   2   6
  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78
   7   6   4   3   7   8   8   7   6   7   1   8   1   5   8   5   6   8   8   7   2   4   8   8   5   8
  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104
   7   4   8   7   1   6   4   1   4   6   2   4   7   8   8   8   1   6   4   8   1   6   6   5   6   6
 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130
   7   7   2   6   7   6   3   2   1   6   5   8   1   7   6   4   8   8   6   1   5   2   8   3   5   4
 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156
```

```
   5   1   3   7   5   8   5   4   4   8   5   4   7   1   5   8   5   6   4   8   5   8   2   7
 989 990 991 992 993 994 995 996 997 998 999 1000
   5   1   6   6   1   2   7   8   4   1   4   1
 [ reached getOption("max.print") -- omitted 266 entries ]

Within cluster sum of squares by cluster:
[1] 2.529937 4.700406 8.449403 1.736631 3.129309 3.195468 2.509610 1.831329
 (between_SS / total_SS =  96.5 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
>
```

```
> subset_data_60_40.test.ct.k8 = CrossTable(subset_data_60_40.test.pred.k8$cluster,subset_data_60_40.test.kmeans.k8$cluster,prop.chisq = TRUE)

   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|
```

Total Observations in Table: 845

subset_data_60_40.test.pred.k8$cluster	subset_data_60_40.test.kmeans.k8$cluster 1	2	3	4	5	6	7	8	Row Total
1	0 7.757 0.000 0.000 0.000	23 46.502 0.333 0.307 0.027	3 7.599 0.043 0.019 0.004	1 6.771 0.014 0.009 0.001	24 12.744 0.348 0.167 0.028	8 1.585 0.116 0.127 0.009	6 1.734 0.087 0.048 0.007	4 0.931 0.058 0.051 0.005	69 0.082
2	13 0.050 0.106 0.137 0.015	6 2.215 0.049 0.080 0.007	32 3.523 0.260 0.203 0.038	19 0.826 0.154 0.179 0.022	22 0.052 0.179 0.153 0.026	6 1.096 0.049 0.095 0.007	17 0.079 0.138 0.136 0.020	8 1.065 0.065 0.101 0.009	123 0.146
3	27 8.277 0.194 0.284 0.032	0 12.337 0.000 0.000 0.000	42 9.861 0.302 0.266 0.050	25 3.281 0.180 0.236 0.030	15 3.186 0.108 0.104 0.018	0 10.363 0.000 0.000 0.000	16 1.012 0.115 0.128 0.019	14 0.078 0.101 0.177 0.017	139 0.164
4	7 3.206 0.058 0.074 0.008	7 1.302 0.058 0.093 0.008	22 0.017 0.182 0.139 0.026	9 2.515 0.074 0.085 0.011	32 6.280 0.264 0.222 0.038	14 2.748 0.116 0.222 0.017	20 0.247 0.165 0.160 0.024	10 0.152 0.083 0.127 0.012	121 0.143
5	1 14.811 0.007 0.011 0.001	27 14.348 0.181 0.360 0.032	6 17.153 0.040 0.038 0.007	5 10.029 0.034 0.047 0.006	49 21.950 0.329 0.340 0.058	35 51.381 0.235 0.556 0.041	22 0.000 0.148 0.176 0.026	4 7.079 0.027 0.051 0.005	149 0.176
6	16 8.758 0.232 0.168 0.019	0 6.124 0.000 0.000 0.000	20 3.905 0.290 0.127 0.024	11 0.635 0.159 0.104 0.013	0 11.759 0.000 0.000 0.000	0 5.144 0.000 0.000 0.000	9 0.143 0.130 0.072 0.011	13 6.649 0.188 0.165 0.015	69 0.082
7	0 2.136 0.000 0.000 0.000	12 63.076 0.632 0.160 0.014	0 3.553 0.000 0.000 0.000	0 2.383 0.000 0.000 0.000	0 3.238 0.000 0.000 0.000	0 1.417 0.000 0.000 0.000	6 3.619 0.316 0.048 0.007	1 0.339 0.053 0.013 0.001	19 0.022
8	31 10.332 0.199 0.326 0.037	0 13.846 0.000 0.000 0.000	33 0.503 0.212 0.209 0.039	36 13.796 0.231 0.340 0.043	2 22.735 0.013 0.014 0.002	0 11.631 0.000 0.000 0.000	29 1.520 0.186 0.232 0.034	25 7.438 0.160 0.316 0.030	156 0.185
Column Total	95 0.112	75 0.089	158 0.187	106 0.125	144 0.170	63 0.075	125 0.148	79 0.093	845

Cluster	subset_data_70_30.test.kmeans.k8	subset_data_70_30.test.pred.k8
1	**1**	**1**
2	**75**	**123**
3	**158**	**139**
4	**106**	**121**
5	**144**	**149**
6	**63**	**69**
7	**125**	**19**
8	**79**	**156**

50-50 SPLIT

```
> #Compare Actual Vs Prediction
> subset_data_50_50.test.pred.k8 = kmeans(subset_data_50_50.test.pred,centers = 8)
> subset_data_50_50.test.pred.k8
K-means clustering with 8 clusters of sizes 110, 155, 176, 187, 172, 154, 34, 68

Cluster means:
      [,1]
1 4.195427
2 3.473623
3 2.284219
4 2.690695
5 3.814802
6 3.111240
7 4.733641
8 1.695791

Clustering vector:
   1   2   5   7   9  10  13  16  17  18  19  20  23  25  27  28  29  30  31  34  36  37  38  39  40  44  48  49  50  52  55  56  57  62  64  67  68  69  71  73  75  76  77  80  81  84
   3   6   4   6   5   3   2   2   2   4   5   4   2   6   6   4   6   4   4   1   3   8   3   3   6   4   3   6   3   8   6   4   3   3   3   3   4   6   3   4   6   2   3   3   4   6
  85  86  89  92  94  96  99 100 101 105 107 115 116 118 119 122 123 124 129 130 131 132 137 140 142 143 144 145 146 147 150 151 152 153 154 155 156 158 160 161 165 166 167 169 170 174
   4   3   3   2   4   5   4   6   6   5   4   2   6   3   3   4   4   6   4   1   8   8   6   3   1   4   1   3   3   2   1   6   5   2   2   4   1   3   6   6   7   5   1   6   5   4
 175 176 177 178 180 181 187 188 189 196 204 207 211 212 213 214 216 219 220 222 223 224 225 226 228 229 230 231 233 234 236 237 241 242 247 248 249 250 251 254 257 258 261 262 263 264
   3   5   3   5   3   8   7   6   7   2   2   3   3   8   3   6   2   8   4   3   2   6   3   1   2   7   1   1   7   7   3   8   3   4   6   4   2   2   6   4   4   4   8   2   3   3
 266 267 268 270 271 272 273 281 282 285 286 287 288 289 291 294 296 297 300 303 304 305 306 308 311 313 314 316 317 319 320 321 325 327 328 329 330 332 333 334 340 341 342 343 345 346
   8   7   7   4   6   8   8   2   3   8   3   3   3   3   3   3   8   4   2   2   8   6   4   3   3   3   8   4   1   3   3   4   6   6   6   2   4   8   6   5   3   6   8   2   2   4
 347 356 357 358 359 363 364 365 366 369 370 371 372 376 378 380 381 382 383 384 385 386 388 389 390 391 393 394 397 399 400 401 403 405 407 411 414 415 417 422 423 424 425 426 427 430
   3   6   6   8   1   7   4   6   2   5   4   3   4   7   3   4   2   6   4   3   3   8   6   3   8   4   6   8   1   2   4   8   1   5   2   4   2   1   1   2   3   5   6   3   4   6
 431 433 438 439 443 444 448 449 450 464 465 466 469 470 472 476 478 479 482 484 486 488 489 491 492 493 496 497 499 500 504 507 508 509 510 512 517 521 522 523 527 528 529 531 532 535
```

```
Within cluster sum of squares by cluster:
[1] 1.514448 1.453878 3.558732 2.210009 1.780324 2.005590 2.048752 3.827695
 (between_SS / total_SS =  97.1 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"         "ifault"
>
```

```
> subset_data_50_50.test.ct.k8 = CrossTable(subset_data_50_50.test.pred.k8$cluster,subset_data_50_50.test.kmeans.k8$cluster,prop.chisq = TRUE)


   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  1056
```

Total Observations in Table: 1056

subset_data_50_50.test.pred.k8$cluster	subset_data_50_50.test.kmeans.k8$cluster 1	2	3	4	5	6	7	8	Row Total
1	40	14	0	6	17	27	2	4	110
	21.956	3.667	13.333	4.537	0.012	25.812	8.212	8.941	
	0.364	0.127	0.000	0.055	0.155	0.245	0.018	0.036	0.104
	0.215	0.173	0.000	0.045	0.107	0.267	0.018	0.026	
	0.038	0.013	0.000	0.006	0.016	0.026	0.002	0.004	
2	35	23	7	10	25	22	11	22	155
	2.171	10.383	7.396	4.753	0.118	3.473	1.964	0.009	
	0.226	0.148	0.045	0.065	0.161	0.142	0.071	0.142	0.147
	0.188	0.284	0.055	0.075	0.157	0.218	0.096	0.144	
	0.033	0.022	0.007	0.009	0.024	0.021	0.010	0.021	
3	3	0	34	35	26	0	43	35	176
	25.290	13.500	7.521	7.184	0.009	16.833	30.316	3.539	
	0.017	0.000	0.193	0.199	0.148	0.000	0.244	0.199	0.167
	0.016	0.000	0.266	0.261	0.164	0.000	0.377	0.229	
	0.003	0.000	0.032	0.033	0.025	0.000	0.041	0.033	
4	22	0	42	28	27	2	28	38	187
	3.632	14.344	16.490	0.769	0.047	14.109	3.023	4.390	
	0.118	0.000	0.225	0.150	0.144	0.011	0.150	0.203	0.177
	0.118	0.000	0.328	0.209	0.170	0.020	0.246	0.248	
	0.021	0.000	0.040	0.027	0.026	0.002	0.027	0.036	
5	68	39	1	7	29	21	2	5	172
	46.926	50.480	18.896	10.071	0.372	1.258	14.784	15.924	
	0.395	0.227	0.006	0.041	0.169	0.122	0.012	0.029	0.163
	0.366	0.481	0.008	0.052	0.182	0.208	0.018	0.033	
	0.064	0.037	0.001	0.007	0.027	0.020	0.002	0.005	
6	17	5	19	29	22	9	19	34	154
	3.779	3.929	0.006	4.578	0.061	2.228	0.339	6.122	
	0.110	0.032	0.123	0.188	0.143	0.058	0.123	0.221	0.146
	0.091	0.062	0.148	0.216	0.138	0.089	0.167	0.222	
	0.016	0.005	0.018	0.027	0.021	0.009	0.018	0.032	
7	1	0	2	1	10	20	0	0	34
	4.156	2.608	1.092	2.546	4.653	86.257	3.670	4.926	
	0.029	0.000	0.059	0.029	0.294	0.588	0.000	0.000	0.032
	0.005	0.000	0.016	0.007	0.063	0.198	0.000	0.000	
	0.001	0.000	0.002	0.001	0.009	0.019	0.000	0.000	
8	0	0	23	18	3	0	9	15	68
	11.977	5.216	26.423	10.178	5.118	6.504	0.375	2.690	
	0.000	0.000	0.338	0.265	0.044	0.000	0.132	0.221	0.064
	0.000	0.000	0.180	0.134	0.019	0.000	0.079	0.098	
	0.000	0.000	0.022	0.017	0.003	0.000	0.009	0.014	
Column Total	186	81	128	134	159	101	114	153	1056
	0.176	0.077	0.121	0.127	0.151	0.096	0.108	0.145	

Cluster	subset_data_70_30.test.kmeans.k8	subset_data_70_30.test.pred.k8
1	**1**	**1**
2	**75**	**123**
3	**158**	**139**
4	**106**	**121**
5	**144**	**149**
6	**63**	**69**
7	**125**	**19**
8	**79**	**156**

Merely tallying the sum of diagonal elements within the table, where both K-means clustering and our forecasts concur in assigning observations to the same cluster, unveils a frequent disparity between our predictions and the K-means clusters. This inconsistency indicates insufficient performance in our forecasting. This disparity underscores the complexity of the relationships between the variables and suggests that a purely linear model may not adequately capture the underlying patterns in the data

To enhance our understanding of the outcomes, we will now examine several performance metrics. To accomplish this, we begin by computing the confusion matrix. Subsequently, we utilize the outcomes from the confusion matrix to calculate our metrics, as illustrated below.

```r
install.packages("caret")
library(caret)
conf_matrix_70_30 <- table(subset_data_70_30.test.pred.k8$cluster, subset_data_70_30.test.kmeans.k8$cluster)

# Calculate True Positives (TP)
TP <- sum(diag(conf_matrix_70_30))

# Calculate False Positives (FP)
FP <- sum(conf_matrix_70_30) - TP

# Calculate False Negatives (FN)
FN <- sum(conf_matrix_70_30) - TP

# Calculate True Negatives (TN)
TN <- sum(conf_matrix_70_30) - FP

# Calculate Precision
precision <- TP / (TP + FP)

# Calculate Recall (Sensitivity)
recall <- TP / (TP + FN)

# Calculate Specificity
specificity <- TN / (TN + FP)

# Calculate F1 Score
f1_score <- 2 * precision * recall / (precision + recall)

# Calculate Error
error <- 1 - (TP + TN) / sum(conf_matrix_70_30)

# Calculate Accuracy
accuracy <- (TP + TN) / sum(conf_matrix_70_30)
```

Following the analysis of all three data splits, we tabulated the results for the following key performance indicators (KPIs): precision and error. Due to the availability of data for only true positives and false positives, these metrics were determined to be the most appropriate for this evaluation.

Split	Precision	Error	TP	FP
70-30%	15.93%	68.13%	101	533
60-40%	17.39%	65.20%	147	698
50-50%	9.09%	81.87%	96	960

Our analysis revealed that the precision rates are low, particularly for the 50-50 training-testing split (9.09%). There's a noticeable improvement with the 60-40 split (17.39%). This suggests that we should investigate the model's attributes further. We might need to remove some attributes to achieve better precision.

CONCLUSION

This project has provided a comprehensive learning experience in several key aspects of data science, particularly in data exploration, clustering, and predictive analysis. Here's a breakdown of the insights we both gained through each phase of the project:

Data Exploration:

- By conducting pairwise plotting and exploring the correlations between attributes, I gained a deeper understanding of the relationships within the dataset.
- Visualizations such as scatter plots and correlation matrices helped to identify potential patterns and dependencies between variables.
- Investigating statistics of the dataset through summary() and describe() functions provided insights into the distribution and summary statistics of the data.

Data Preparation:

- The process of subsetting the data and selecting the most relevant attributes involved critical decision-making based on correlation analysis and domain knowledge.
- Translating alphanumeric values into numeric values required thoughtful mapping strategies to ensure accurate representation of categorical variables.

Clustering:

- Applying clustering algorithms to the dataset allowed for the identification of natural groupings or clusters within the data.
- Utilizing methods like k-means clustering and visualizing the clusters using tools like factoextra:fviz_cluster() facilitated the interpretation and understanding of the data structure.

Prediction:

- Implementing predictive models using techniques such as Generalized Linear Models (GLM) provided insights into the predictive power of the selected features.
- Evaluating model performance through measures like accuracy, false positives, etc., offered a quantitative assessment of the predictive capabilities of the models.

This project has demonstrated the iterative and interdisciplinary nature of data science. It required a combination of statistical knowledge, programming skills, and domain expertise to effectively explore, analyze, and derive insights from the dataset. It also highlighted the importance of data preprocessing, feature selection, and model evaluation in the predictive modeling process. Through hands-on experience with real-world data, we've gained valuable insights into the practical application of data science techniques and their implications for decision-making and problem-solving in various domains.