Project 1

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3

## Warning: package 'tidyr' was built under R version 4.4.2

## Warning: package 'readr' was built under R version 4.4.2

## Warning: package 'purrr' was built under R version 4.4.2

## Warning: package 'dplyr' was built under R version 4.4.2

## Warning: package 'forcats' was built under R version 4.4.2

## Warning: package 'lubridate' was built under R version 4.4.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
data <- read.csv("ObesityDataSet.csv")
```

```
head(data)
```

```
##   Gender Age Height Weight family_history_with_overweight FAVC FCVC NCP
## 1 Female  21   1.62   64.0                              yes   no    2   3
## 2 Female  21   1.52   56.0                              yes   no    3   3
## 3   Male  23   1.80   77.0                              yes   no    2   3
## 4   Male  27   1.80   87.0                               no   no    3   3
## 5   Male  22   1.78   89.8                               no   no    2   1
## 6   Male  29   1.62   53.0                               no  yes    2   3
##        CAEC SMOKE CH2O SCC FAF TUE       CALC                MTRANS
## 1 Sometimes    no    2  no   0   1         no Public_Transportation
## 2 Sometimes   yes    3 yes   3   0  Sometimes Public_Transportation
## 3 Sometimes    no    2  no   2   1 Frequently Public_Transportation
```

```
## 4 Sometimes    no    2  no   2   0 Frequently               Walking
## 5 Sometimes    no    2  no   0   0  Sometimes Public_Transportation
## 6 Sometimes    no    2  no   0   0  Sometimes            Automobile
##            NObeyesdad
## 1       Normal_Weight
## 2       Normal_Weight
## 3       Normal_Weight
## 4  Overweight_Level_I
## 5 Overweight_Level_II
## 6       Normal_Weight
```

```
str(data)
```

```
## 'data.frame':    2111 obs. of  17 variables:
##  $ Gender                         : chr  "Female" "Female" "Male" "Male" ...
##  $ Age                            : num  21 21 23 27 22 29 23 22 24 22 ...
##  $ Height                         : num  1.62 1.52 1.8 1.8 1.78 1.62 1.5 1.64 1.78 1.72 ...
##  $ Weight                         : num  64 56 77 87 89.8 53 55 53 64 68 ...
##  $ family_history_with_overweight: chr  "yes" "yes" "yes" "no" ...
##  $ FAVC                           : chr  "no" "no" "no" "no" ...
##  $ FCVC                           : num  2 3 2 3 2 2 3 2 3 2 ...
##  $ NCP                            : num  3 3 3 3 1 3 3 3 3 3 ...
##  $ CAEC                           : chr  "Sometimes" "Sometimes" "Sometimes" "Sometimes" ...
##  $ SMOKE                          : chr  "no" "yes" "no" "no" ...
##  $ CH2O                           : num  2 3 2 2 2 2 2 2 2 2 ...
##  $ SCC                            : chr  "no" "yes" "no" "no" ...
##  $ FAF                            : num  0 3 2 2 0 0 1 3 1 1 ...
##  $ TUE                            : num  1 0 1 0 0 0 0 0 1 1 ...
##  $ CALC                           : chr  "no" "Sometimes" "Frequently" "Frequently" ...
##  $ MTRANS                         : chr  "Public_Transportation" "Public_Transportation" "Public_Trans
##  $ NObeyesdad                     : chr  "Normal_Weight" "Normal_Weight" "Normal_Weight" "Overweight_
```

```
summary(data)
```

```
##     Gender                Age              Height           Weight
##  Length:2111        Min.   :14.00   Min.   :1.450   Min.   : 39.00
##  Class :character   1st Qu.:19.95   1st Qu.:1.630   1st Qu.: 65.47
##  Mode  :character   Median :22.78   Median :1.700   Median : 83.00
##                     Mean   :24.31   Mean   :1.702   Mean   : 86.59
##                     3rd Qu.:26.00   3rd Qu.:1.768   3rd Qu.:107.43
##                     Max.   :61.00   Max.   :1.980   Max.   :173.00
##  family_history_with_overweight     FAVC                FCVC
##  Length:2111                    Length:2111        Min.   :1.000
##  Class :character               Class :character   1st Qu.:2.000
##  Mode  :character               Mode  :character   Median :2.386
##                                                    Mean   :2.419
##                                                    3rd Qu.:3.000
##                                                    Max.   :3.000
##       NCP             CAEC              SMOKE                CH2O
##  Min.   :1.000   Length:2111        Length:2111        Min.   :1.000
##  1st Qu.:2.659   Class :character   Class :character   1st Qu.:1.585
##  Median :3.000   Mode  :character   Mode  :character   Median :2.000
##  Mean   :2.686                                         Mean   :2.008
```

```
##  3rd Qu.:3.000                                          3rd Qu.:2.477
##  Max.   :4.000                                          Max.   :3.000
##      SCC                 FAF              TUE             CALC
##  Length:2111        Min.   :0.0000   Min.   :0.0000   Length:2111
##  Class :character   1st Qu.:0.1245   1st Qu.:0.0000   Class :character
##  Mode  :character   Median :1.0000   Median :0.6253   Mode  :character
##                     Mean   :1.0103   Mean   :0.6579
##                     3rd Qu.:1.6667   3rd Qu.:1.0000
##                     Max.   :3.0000   Max.   :2.0000
##     MTRANS            NObeyesdad
##  Length:2111        Length:2111
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

Data Cleaning and EDA

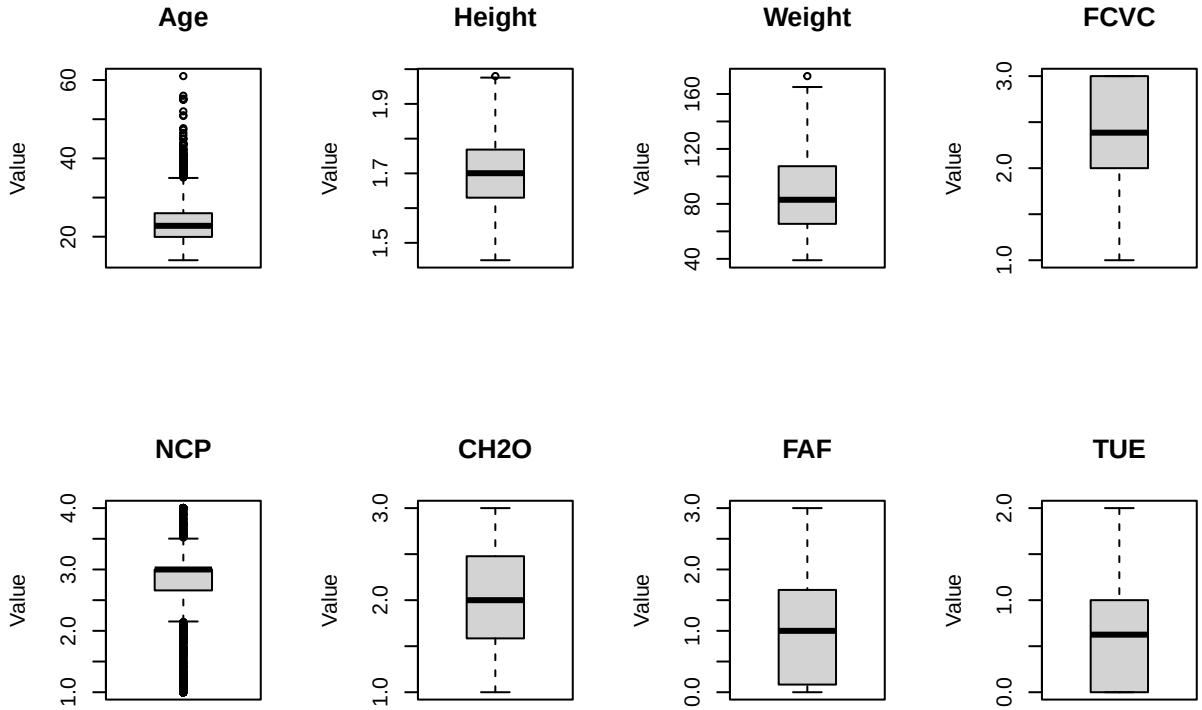
```
cat_cols <- c("Gender", "family_history_with_overweight", "FAVC", "CAEC", "SMOKE",
              "SCC", "CALC", "MTRANS", "NObeyesdad")
data[cat_cols] <-lapply(data[cat_cols], factor)

missing_values_check <- colSums(is.na(data))
print(missing_values_check)
```

```
##                          Gender                             Age
##                               0                               0
##                          Height                          Weight
##                               0                               0
## family_history_with_overweight                            FAVC
##                               0                               0
##                            FCVC                             NCP
##                               0                               0
##                            CAEC                           SMOKE
##                               0                               0
##                            CH2O                             SCC
##                               0                               0
##                             FAF                            TUE
##                               0                               0
##                            CALC                          MTRANS
##                               0                               0
##                      NObeyesdad
##                               0
```

Outlier Detection and Scaling

```
num_cols <- c("Age", "Height", "Weight", "FCVC", "NCP", "CH2O", "FAF", "TUE")

par(mfrow = c(2, 4))
```

```r
for (col in num_cols) {
  boxplot(data[[col]], main = col, ylab = "Value")
}
```



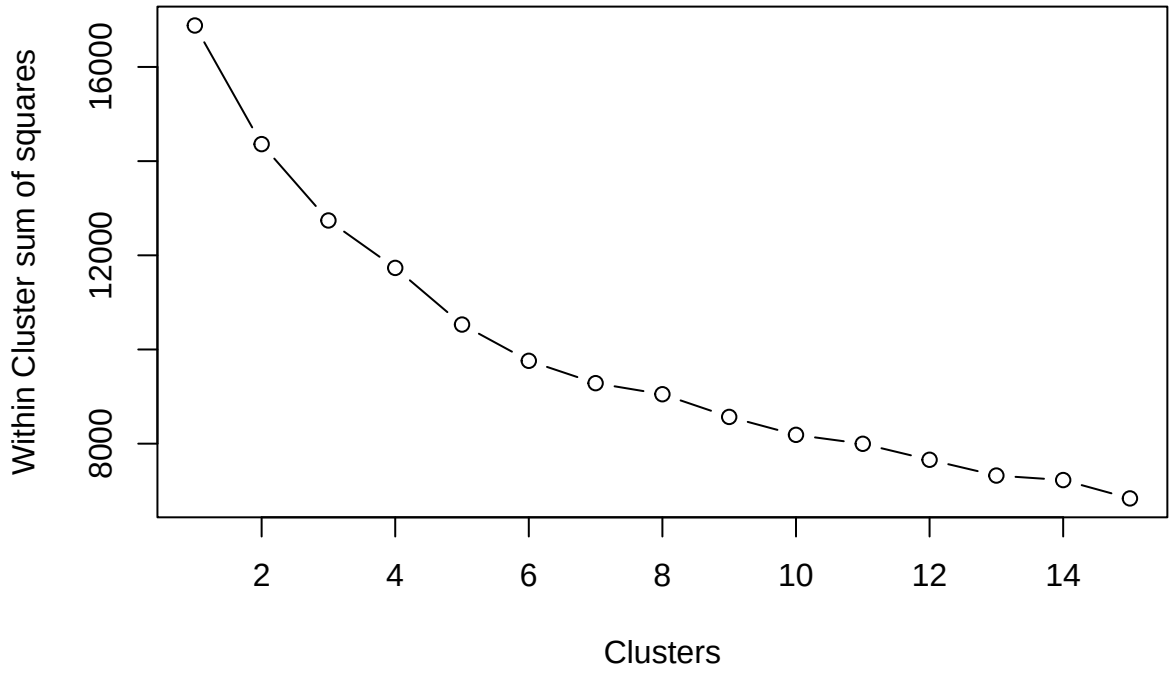

```r
par(mfrow = c(1, 1))
```

```r
data[num_cols] <- scale(data[num_cols])
summary(data[num_cols])
```

```
##       Age               Height             Weight             FCVC         
##  Min.   :-1.6251   Min.   :-2.69737   Min.   :-1.8169   Min.   :-2.65775  
##  1st Qu.:-0.6879   1st Qu.:-0.76821   1st Qu.:-0.8061   1st Qu.:-0.78483  
##  Median :-0.2418   Median :-0.01263   Median :-0.1369   Median :-0.06282  
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000  
##  3rd Qu.: 0.2659   3rd Qu.: 0.71579   3rd Qu.: 0.7959   3rd Qu.: 1.08808  
##  Max.   : 5.7812   Max.   : 2.98294   Max.   : 3.2994   Max.   : 1.08808  
##       NCP               CH2O               FAF               TUE         
##  Min.   :-2.16651   Min.   :-1.64452   Min.   :-1.18776   Min.   :-1.0804  
##  1st Qu.:-0.03456   1st Qu.:-0.69043   1st Qu.:-1.04138   1st Qu.:-1.0804  
##  Median : 0.40406   Median :-0.01307   Median :-0.01211   Median :-0.0534  
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000  
##  3rd Qu.: 0.40406   3rd Qu.: 0.76581   3rd Qu.: 0.77167   3rd Qu.: 0.5619  
##  Max.   : 1.68934   Max.   : 1.61838   Max.   : 2.33920   Max.   : 2.2041
```

K-means clustering

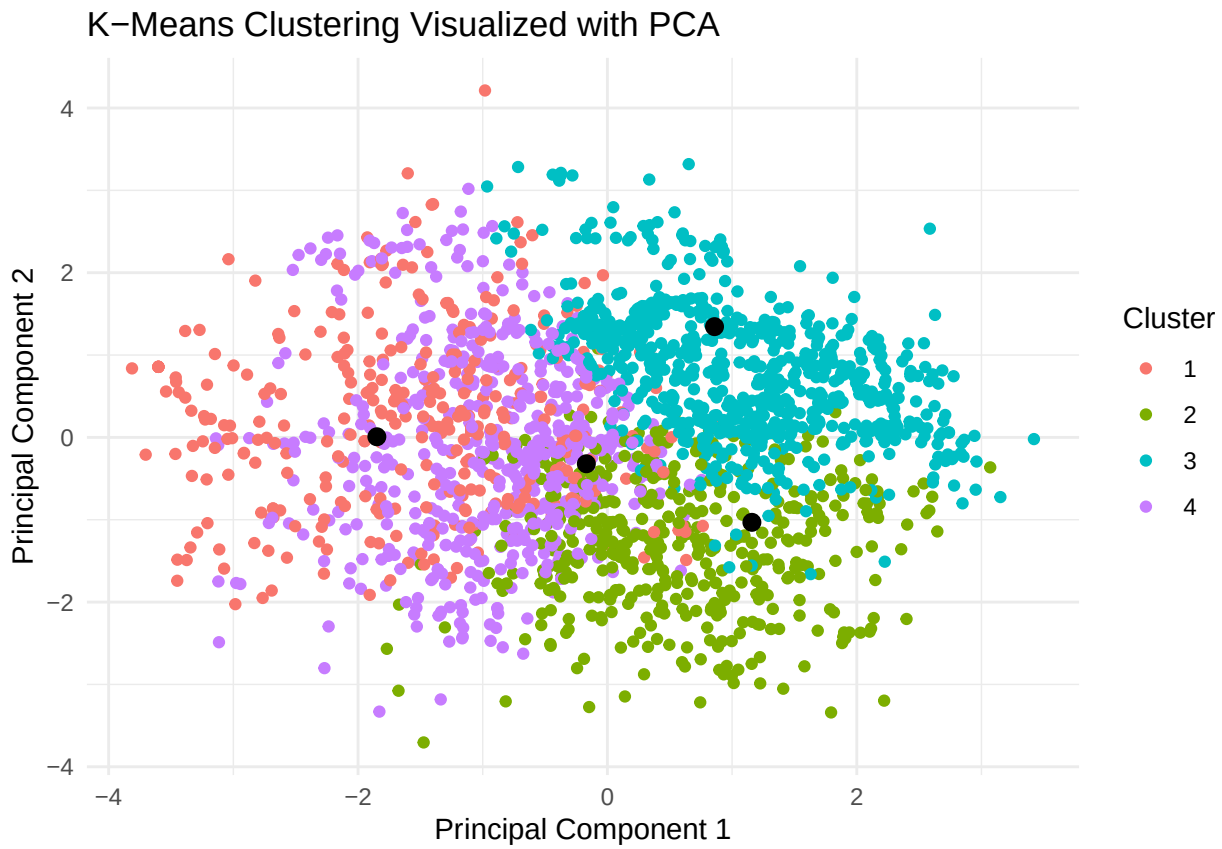
```
scaled_num_col_data <- data[num_cols]

wss <- (nrow(scaled_num_col_data) - 1) * sum(apply(scaled_num_col_data, 2, var))
for (i in 2:15) {
  wss[i] <- sum(kmeans(scaled_num_col_data, centers =i)$withinss)
}

plot(1:15, wss, type = "b", xlab = "Clusters", ylab = "Within Cluster sum of squares")
```




```
set.seed(123)

k <- 4
kmeans_res <- kmeans(scaled_num_col_data, centers = k)

data$cluster <- as.factor(kmeans_res$cluster)

table(data$cluster)
```

```
##
##   1   2   3   4
## 356 446 750 559
```

```r
pca_data <- data[, num_cols]
pca_data <- scale(pca_data)

pca_result <- prcomp(pca_data)

pca_plot_data <- as.data.frame(pca_result$x[, 1:2])
pca_plot_data$cluster <- as.factor(kmeans_res$cluster)

pca_centers <- prcomp(kmeans_res$centers)$x[, 1:2]

ggplot(pca_plot_data, aes(x = PC1, y = PC2, color = cluster)) +
  geom_point() +
  labs(
    title = "K-Means Clustering Visualized with PCA",
    x = "Principal Component 1",
    y = "Principal Component 2",
    color = "Cluster"
  ) + theme_minimal() +
  geom_point(data = as.data.frame(pca_centers),
             aes(x = PC1, y = PC2),
             color = "black",
             size = 3,
             shape = 16)
```



Cluster Interpretation

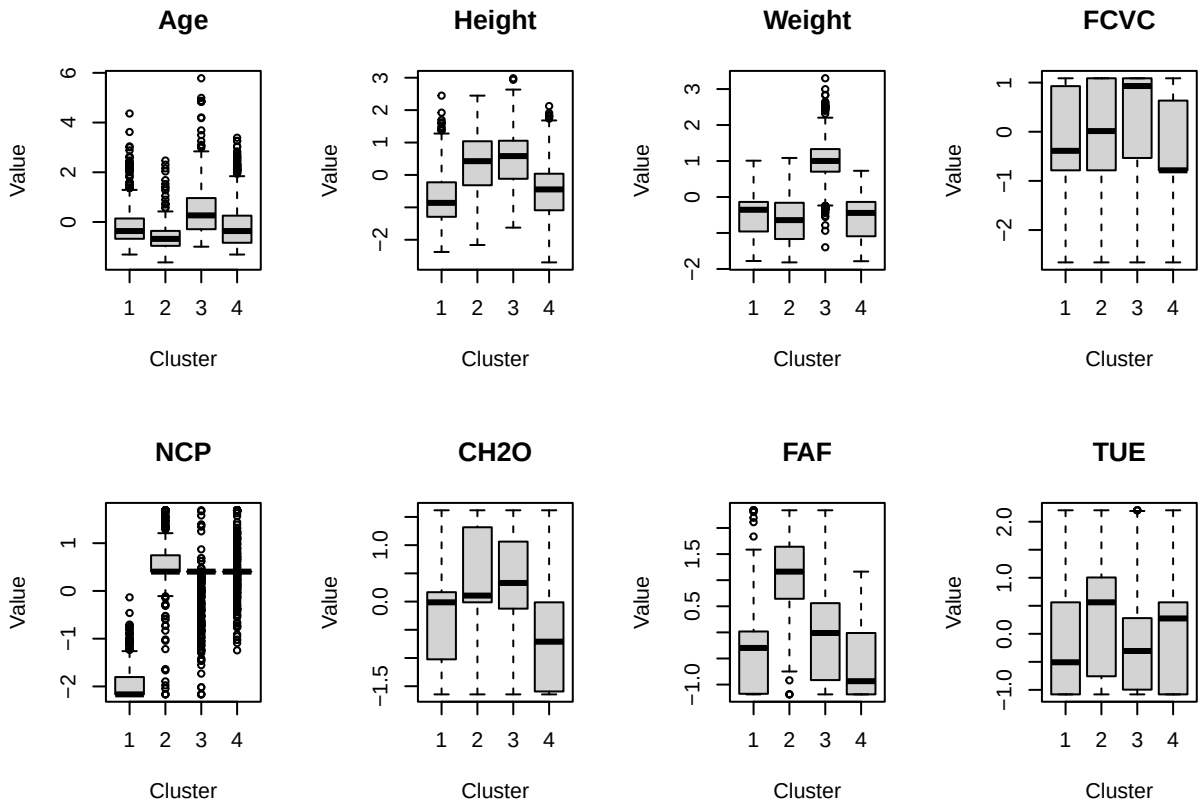
```
library(dplyr)
cluster_summary <- data %>%
  group_by(cluster) %>%
  summarise_all(mean)
```

```
## Warning: There were 36 warnings in `summarise()`.
## The first warning was:
## i In argument: `Gender = (function (x, ...) ...`.
## i In group 1: `cluster = 1`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## i Run `dplyr::last_dplyr_warnings()` to see the 35 remaining warnings.
```

```
print(cluster_summary)
```

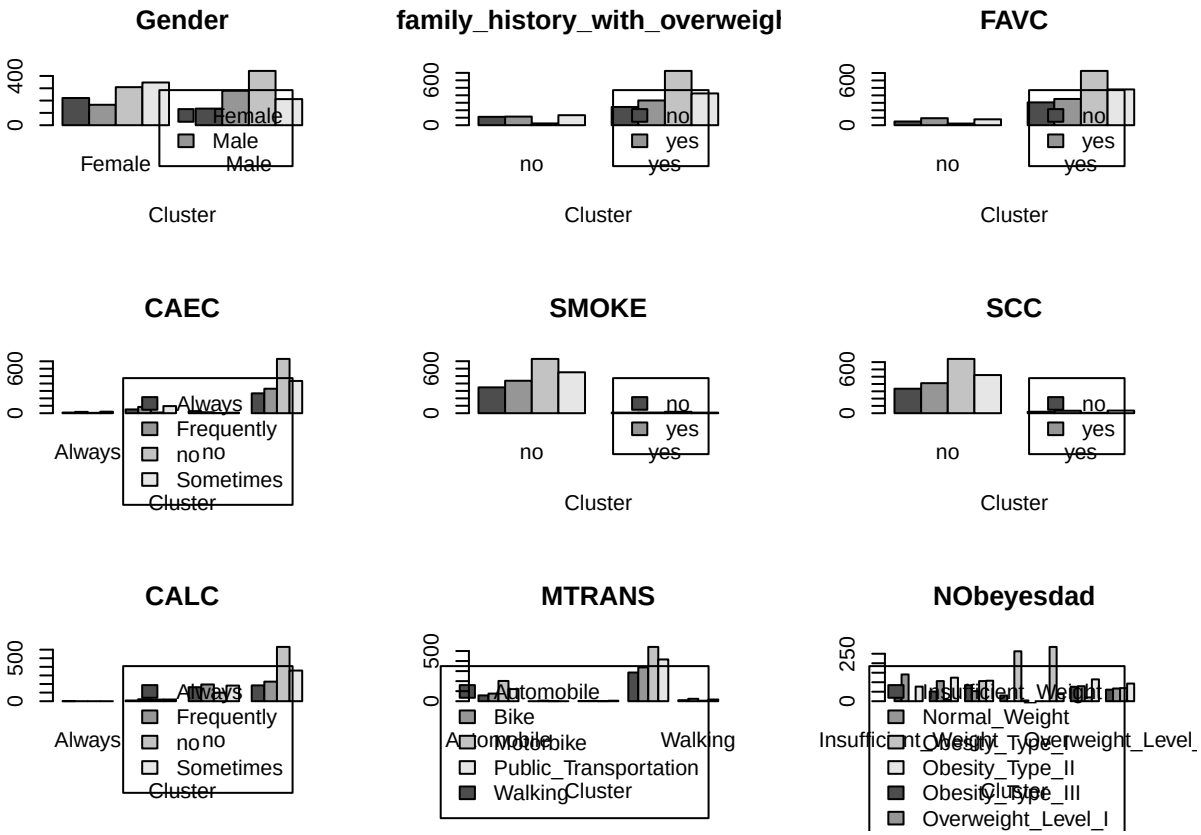
```
## # A tibble: 4 x 18
##   cluster Gender     Age Height Weight family_history_with_over~1  FAVC     FCVC
##   <fct>    <dbl>   <dbl>  <dbl>  <dbl>                      <dbl> <dbl>    <dbl>
## 1 1           NA -0.0651 -0.684 -0.520                        NA    NA -0.0752
## 2 2           NA -0.579   0.301 -0.643                        NA    NA  0.00768
## 3 3           NA  0.428   0.512  1.03                         NA    NA  0.333
## 4 4           NA -0.0706 -0.492 -0.536                        NA    NA -0.405
## # i abbreviated name: 1: family_history_with_overweight
## # i 10 more variables: NCP <dbl>, CAEC <dbl>, SMOKE <dbl>, CH2O <dbl>,
## #   SCC <dbl>, FAF <dbl>, TUE <dbl>, CALC <dbl>, MTRANS <dbl>, NObeyesdad <dbl>
```

```
num_cols <- c("Age", "Height", "Weight", "FCVC", "NCP", "CH2O", "FAF", "TUE")

par(mfrow = c(2, 4))
for (col in num_cols) {
  boxplot(data[[col]] ~ data$cluster, main = col, ylab = "Value", xlab = "Cluster")
}
```




```
par(mfrow = c(1, 1))
```

```
cat_cols <- c("Gender", "family_history_with_overweight", "FAVC", "CAEC", "SMOKE",
              "SCC", "CALC", "MTRANS", "NObeyesdad")

par(mfrow = c(3, 3))
for (col in cat_cols) {
  counts <- table(data$cluster, data[[col]])
  barplot(counts, beside = TRUE, main = col, legend.text = colnames(counts),
          xlab = "Cluster")
}
```




```
par(mfrow = c(1, 1))
```

Feature Importance (using Random Forest)

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.4.2
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

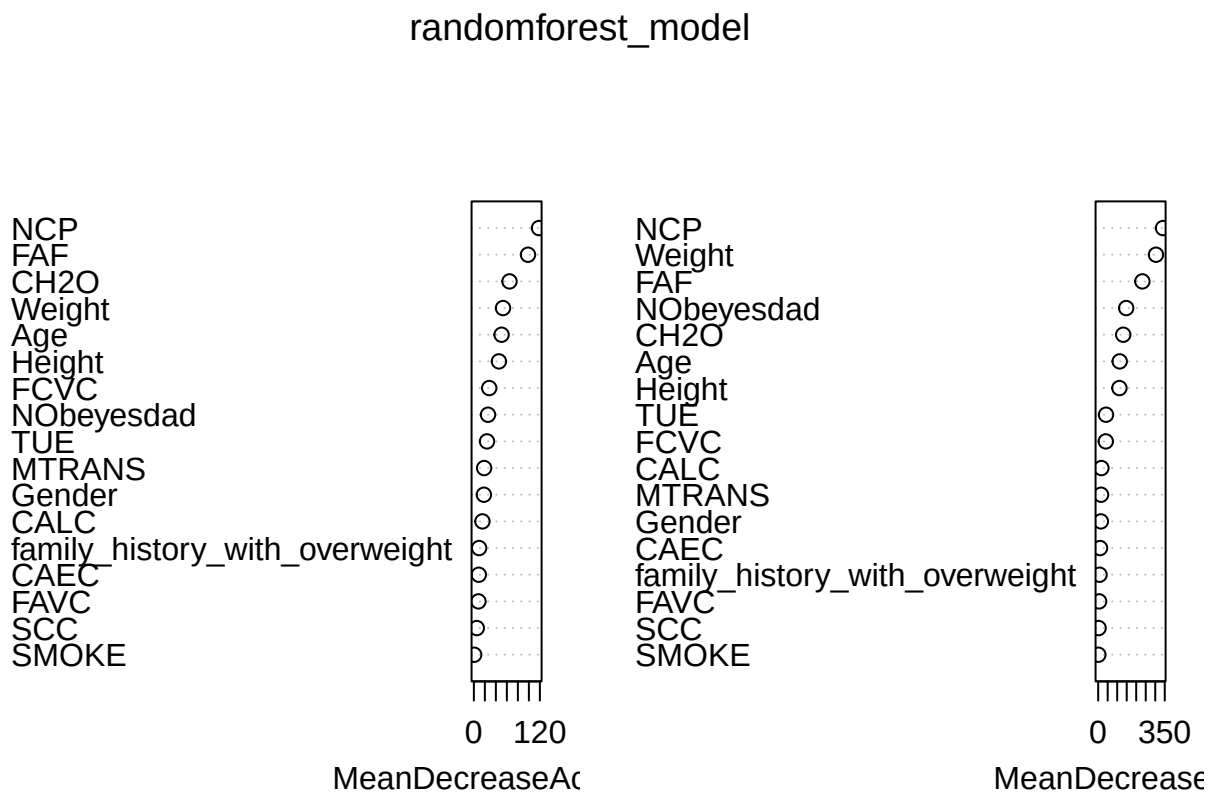
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
data$cluster <- as.factor(data$cluster)
randomforest_data <- data[, c(num_cols, cat_cols, "cluster")]
```

```
randomforest_model <- randomForest(cluster ~ ., data = randomforest_data, importance = TRUE)

imp_scores <- importance(randomforest_model)
```

```
varImpPlot(randomforest_model)
```




```
print(imp_scores)
```

```
##                                      1          2        3          4
## Age                          14.4541444  42.3669653 41.432221 23.2448460
## Height                       22.6486169  26.3983523 25.216641 44.9571939
## Weight                       26.8933824  38.0201446 47.722703 42.7881940
## FCVC                          8.9318408  13.2755409 11.774994 33.0223172
## NCP                         143.7249692  57.3772885 31.535331 99.4354220
## CH2O                         17.3472568  40.1634956 32.455820 67.9923730
## FAF                          19.2555675 101.0517808 26.026025 90.5407771
## TUE                          12.5991878  16.1565362 16.947604 12.8290257
## Gender                        8.4662437   9.5077989 12.658757 14.2677664
## family_history_with_overweight  7.6063405   3.2770248  6.480076  7.3954172
## FAVC                          3.2808345   1.5919171  5.107895  6.2916432
## CAEC                          5.1637257   0.3894613  7.850697  2.9636308
```

```
## SMOKE                            -1.4142362  -0.7357115  1.243752  0.8463491
## SCC                               0.3916292   0.2545040  4.648398  3.4920204
## CALC                             11.6893455   9.0166695 12.521075  9.8697873
## MTRANS                            6.7273762  11.6903927 17.542700  5.9959360
## NObeyesdad                       14.9089200  24.4666379 20.633429 22.3932651
##                                 MeanDecreaseAccuracy MeanDecreaseGini
## Age                                        50.357931      113.1505751
## Height                                     45.573253      111.6665686
## Weight                                     52.997102      304.0226360
## FCVC                                       27.923876       39.9343993
## NCP                                       118.426787      340.7919646
## CH2O                                       64.831160      131.7002596
## FAF                                        98.436124      232.3653055
## TUE                                        23.888866       41.2921605
## Gender                                     18.217839       14.2634435
## family_history_with_overweight              9.559531        7.2063212
## FAVC                                        8.343257        5.0437597
## CAEC                                        8.985868        9.6889545
## SMOKE                                       0.578690        0.8925312
## SCC                                         5.313991        2.2537126
## CALC                                       15.585118       16.8460607
## MTRANS                                     18.664438       15.1380550
## NObeyesdad                                 25.760870      147.3942747
```

Predictive Modeling (Random Forest Classifier)

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.2
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
data$cluster <- as.factor(data$cluster)
randomforest_data <- data[, c("TUE", "Age", "Weight", "Height", "FAF", "FCVC",
                              "NObeyesdad", "cluster")]
```

```
set.seed(123)
train_index <- createDataPartition(randomforest_data$cluster, p = 0.8, list = FALSE)
train_data <- randomforest_data[train_index, ]
test_data <- randomforest_data[-train_index, ]
```

```r
set.seed(123)
randomforest_model <- randomForest(cluster ~ ., data = train_data)

predictions <- predict(randomforest_model, newdata = test_data)
```

```r
confusionMatrix(predictions, test_data$cluster)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3   4
##          1  51   1   2   8
##          2   5  82   3   5
##          3   6   1 139   7
##          4   9   5   6  91
##
## Overall Statistics
##
##                Accuracy : 0.8622
##                  95% CI : (0.8256, 0.8937)
##     No Information Rate : 0.3563
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8109
##
##  Mcnemar's Test P-Value : 0.4457
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity            0.7183   0.9213   0.9267   0.8198
## Specificity            0.9686   0.9608   0.9483   0.9355
## Pos Pred Value         0.8226   0.8632   0.9085   0.8198
## Neg Pred Value         0.9443   0.9785   0.9590   0.9355
## Prevalence             0.1686   0.2114   0.3563   0.2637
## Detection Rate         0.1211   0.1948   0.3302   0.2162
## Detection Prevalence   0.1473   0.2257   0.3634   0.2637
## Balanced Accuracy      0.8434   0.9411   0.9375   0.8777
```

Predictive Modeling (Gradient Boosting Machine)

```r
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.4.3
```

```
## Loaded gbm 2.2.2
```

```
## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.c
```

```r
data$cluster <- as.factor(data$cluster)
gbm_data <- data[, c("TUE", "Age", "Weight", "Height", "FAF", "FCVC", "NObeyesdad",
                     "cluster")]

set.seed(123)
train_index <- createDataPartition(gbm_data$cluster, p = 0.8, list = FALSE)
train_data <- gbm_data[train_index, ]
test_data <- gbm_data[-train_index, ]
```

```r
set.seed(123)
gbm_model <- train(
  cluster ~ .,
  data = train_data,
  method = "gbm",
  trControl = trainControl(method = "cv", number = 10),
  verbose = FALSE
)
```

```r
set.seed(123)
predictions <- predict(gbm_model, newdata = test_data)

confusionMatrix(predictions, test_data$cluster)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3   4
##          1  47   2   3  10
##          2   8  82   3   5
##          3   6   1 139   6
##          4  10   4   5  90
##
## Overall Statistics
##
##                Accuracy : 0.8504
##                  95% CI : (0.8126, 0.8831)
##     No Information Rate : 0.3563
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7947
##
##  Mcnemar's Test P-Value : 0.4457
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity            0.6620   0.9213   0.9267   0.8108
## Specificity            0.9571   0.9518   0.9520   0.9387
## Pos Pred Value         0.7581   0.8367   0.9145   0.8257
## Neg Pred Value         0.9331   0.9783   0.9591   0.9327
## Prevalence             0.1686   0.2114   0.3563   0.2637
## Detection Rate         0.1116   0.1948   0.3302   0.2138
## Detection Prevalence   0.1473   0.2328   0.3610   0.2589
```

```
## Balanced Accuracy      0.8096   0.9366   0.9393   0.8748
```

Predictive Modeling (Support Vector Machine)

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.4.2
```

```
data$cluster <- as.factor(data$cluster)
svm_data <- data[, c("TUE", "Age", "Weight", "Height", "FAF", "FCVC", "NObeyesdad",
                     "cluster")]

set.seed(123)
train_index <- createDataPartition(svm_data$cluster, p = 0.8, list = FALSE)
train_data <- svm_data[train_index, ]
test_data <- svm_data[-train_index, ]
```

```
svm_model <- train(
  cluster ~ .,
  data = train_data,
  method = "svmLinear",
  trControl = trainControl(method = "cv", number = 10),
  preProcess = c("center", "scale")
)
```

```
predictions <- predict(svm_model, newdata = test_data)

confusionMatrix(predictions, test_data$cluster)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3   4
##          1  16   3   4   9
##          2   9  80   4   5
##          3   7   0 138   8
##          4  39   6   4  89
##
## Overall Statistics
##
##                Accuracy : 0.7672
##                  95% CI : (0.7239, 0.8068)
##     No Information Rate : 0.3563
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6776
##
##  Mcnemar's Test P-Value : 9.427e-05
##
## Statistics by Class:
```

```
##
##                      Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity           0.22535   0.8989   0.9200   0.8018
## Specificity           0.95429   0.9458   0.9446   0.8419
## Pos Pred Value        0.50000   0.8163   0.9020   0.6449
## Neg Pred Value        0.85861   0.9721   0.9552   0.9223
## Prevalence            0.16865   0.2114   0.3563   0.2637
## Detection Rate        0.03800   0.1900   0.3278   0.2114
## Detection Prevalence  0.07601   0.2328   0.3634   0.3278
## Balanced Accuracy     0.58982   0.9223   0.9323   0.8219
```

Hyperparameter Tuning for Random Forest

```
data$cluster <- as.factor(data$cluster)
randomforest_data <- data[, c("TUE", "Age", "Weight", "Height", "FAF", "FCVC",
                              "NObeyesdad", "cluster")]

set.seed(123)
train_index <- createDataPartition(randomforest_data$cluster, p = 0.8, list = FALSE)
train_data <- randomforest_data[train_index, ]
test_data <- randomforest_data[-train_index, ]

randomforest_grid <- expand.grid(mtry = c(2, 3, 4, 5, 6, 7))
```

```
rf_tuned_model <- train(
  cluster ~ .,
  data = train_data,
  method = "rf",
  trControl = trainControl(method = "cv", number = 10),
  tuneGrid = randomforest_grid
)
```

```
predictions <- predict(rf_tuned_model, newdata = test_data)

confusionMatrix(predictions, test_data$cluster)
```

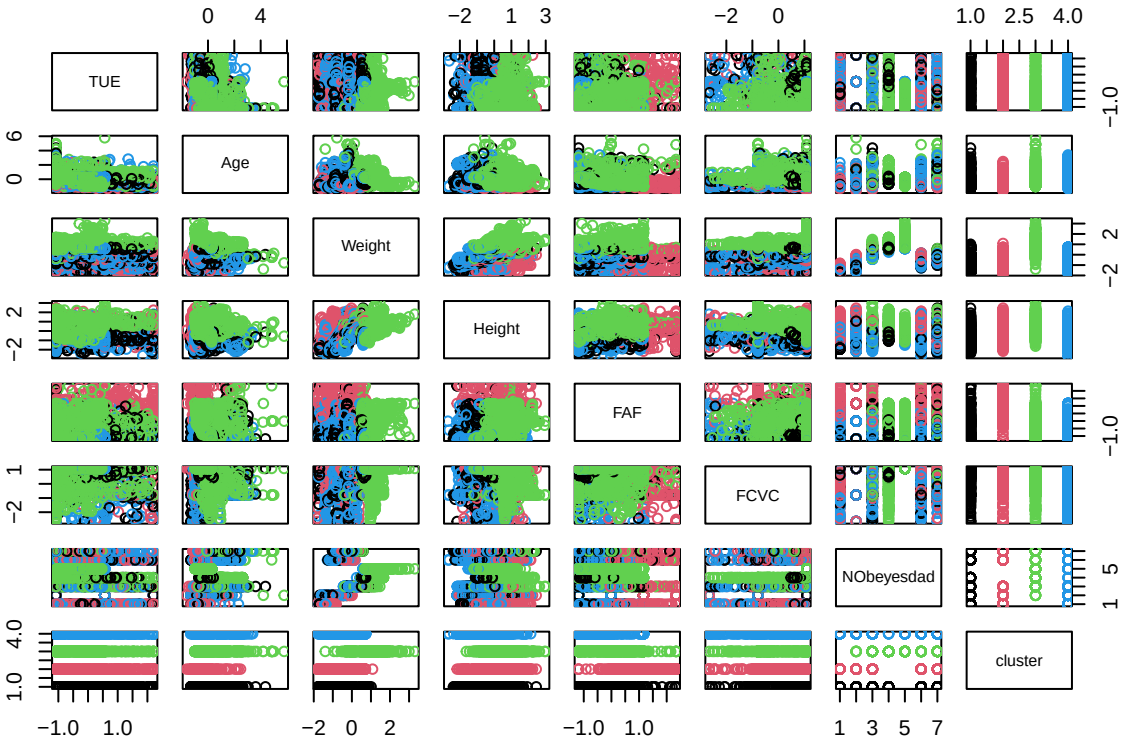
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3   4
##          1  51   1   1  11
##          2   5  82   3   5
##          3   6   1 139   4
##          4   9   5   7  91
##
## Overall Statistics
##
##                Accuracy : 0.8622
##                  95% CI : (0.8256, 0.8937)
##     No Information Rate : 0.3563
##     P-Value [Acc > NIR] : <2e-16
```

```
##
##                   Kappa : 0.8112
##
##  Mcnemar's Test P-Value : 0.2199
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity            0.7183   0.9213   0.9267   0.8198
## Specificity            0.9629   0.9608   0.9594   0.9323
## Pos Pred Value         0.7969   0.8632   0.9267   0.8125
## Neg Pred Value         0.9440   0.9785   0.9594   0.9353
## Prevalence             0.1686   0.2114   0.3563   0.2637
## Detection Rate         0.1211   0.1948   0.3302   0.2162
## Detection Prevalence   0.1520   0.2257   0.3563   0.2660
## Balanced Accuracy      0.8406   0.9411   0.9430   0.8760
```

```
print(rf_tuned_model$bestTune)
```

```
##   mtry
## 6    7
```

```
library(ggplot2)

important_features <- c("TUE", "Age", "Weight", "Height", "FAF", "FCVC", "NObeyesdad")

pairs(data[, c(important_features, "cluster")], col = data$cluster)
```


TUE
Age
Weight
Height
FAF
FCVC
NObeyesdad
cluster