DA 5030 Final Project

Jordan Lian

4/26/2021

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.4

## -- Attaching packages -----------------------------------------------------------------------

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 4.0.4

## -- Conflicts ----------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

https://www.datascience-pm.com/crisp-dm-2/ – An overall CRISPR-DM process that I used as a reference/guide for this project.

1. Business understanding - What does the business need?

The business needs a model(S) that can predict obesity levels based off of certain factors. To achieve this objective, we will require the appropriate data, and some predictive modeling.

2. Data understanding - What data do we have / need? Is it clean?

The data is from the following URL: https://archive.ics.uci.edu/ml/datasets/Estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition+

The citation is below:

Palechor, F. M., & de la Hoz Manotas, A. (2019). Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico. Data in Brief, 104344.

The following URL provides a thorough data description: https://www.sciencedirect.com/science/article/pii/S2352340919306985?via%3Dihub

Load Data

```r
origin <- read_csv("ObesityDataSet_raw_and_data_sinthetic.csv")
```

```
## Parsed with column specification:
## cols(
##   Gender = col_character(),
##   Age = col_double(),
##   Height = col_double(),
##   Weight = col_double(),
##   family_history_with_overweight = col_character(),
##   FAVC = col_character(),
##   FCVC = col_double(),
##   NCP = col_double(),
##   CAEC = col_character(),
##   SMOKE = col_character(),
##   CH2O = col_double(),
##   SCC = col_character(),
##   FAF = col_double(),
##   TUE = col_double(),
##   CALC = col_character(),
##   MTRANS = col_character(),
##   NObeyesdad = col_character()
## )
```

```r
obesity <- origin
```

Explore Data

It is clear that weight and height have the strongest correlation, although the other numerical values do not share any correlation as strong as the one that exists between weight and height. Perhaps it is time that we include the categorical variables in our analysis.

```r
# Count Obesity Types
table(obesity$NObeyesdad)
```

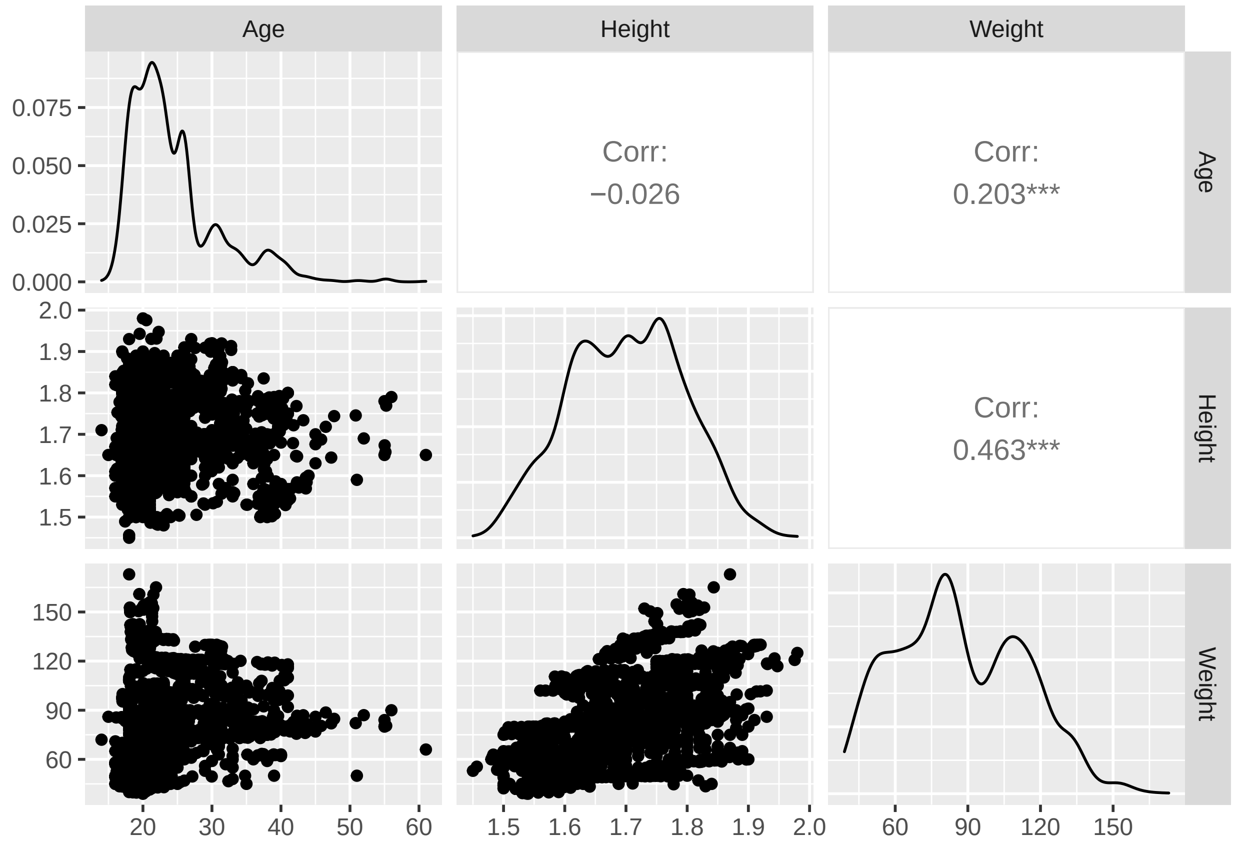
Correlation Plots

```
##
## Insufficient_Weight       Normal_Weight      Obesity_Type_I     Obesity_Type_II
##                 272                 287                 351                 297
##    Obesity_Type_III  Overweight_Level_I Overweight_Level_II
##                 324                 290                 290
```

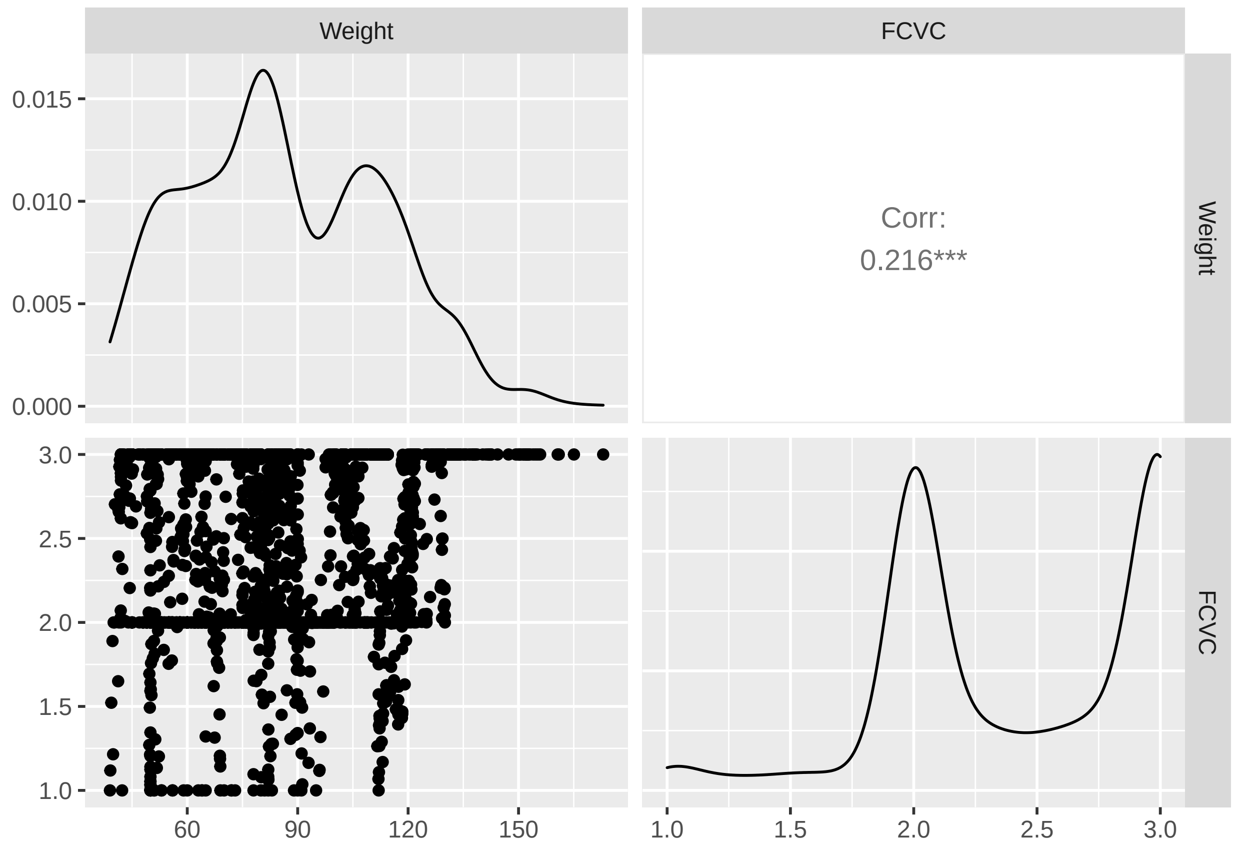
```r
# Age vs Height vs Weight
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.0.3
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

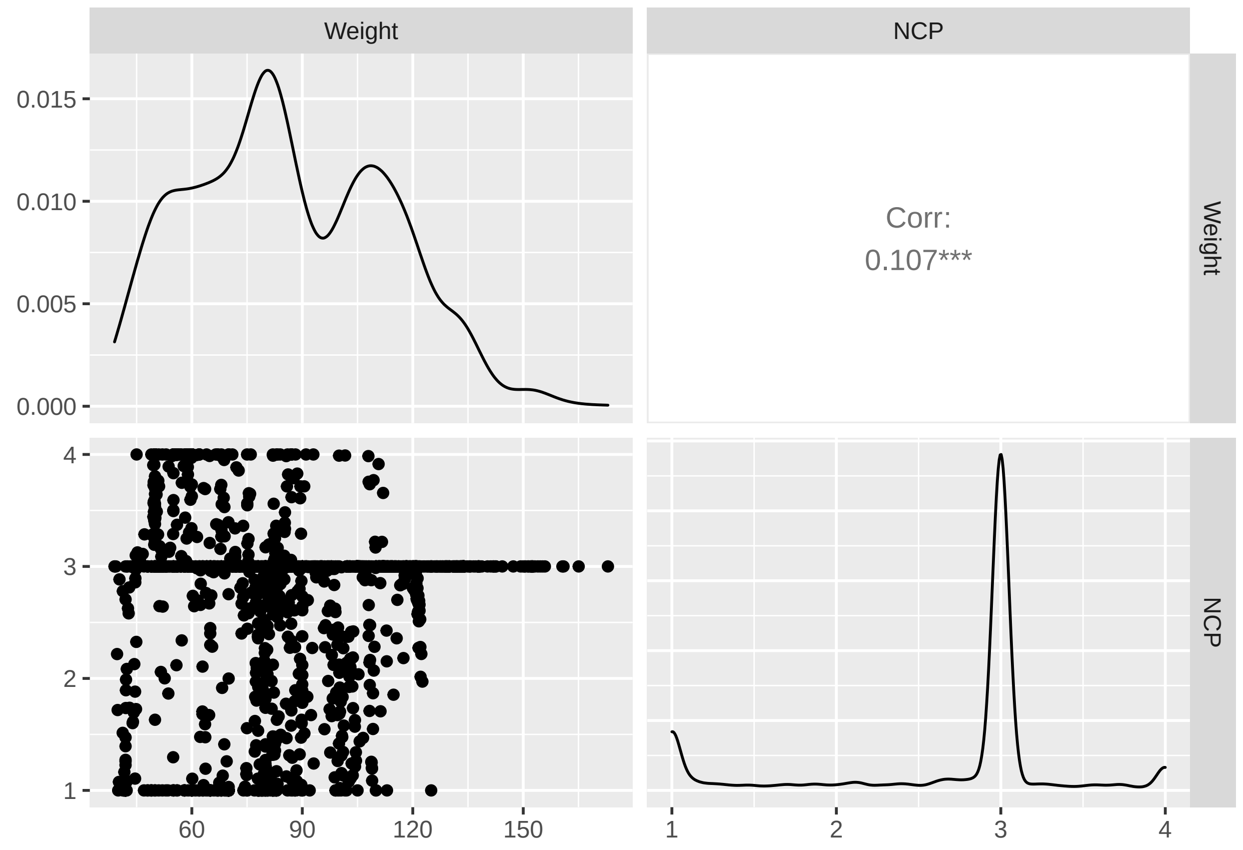
```r
ggpairs(obesity[2:4])
```



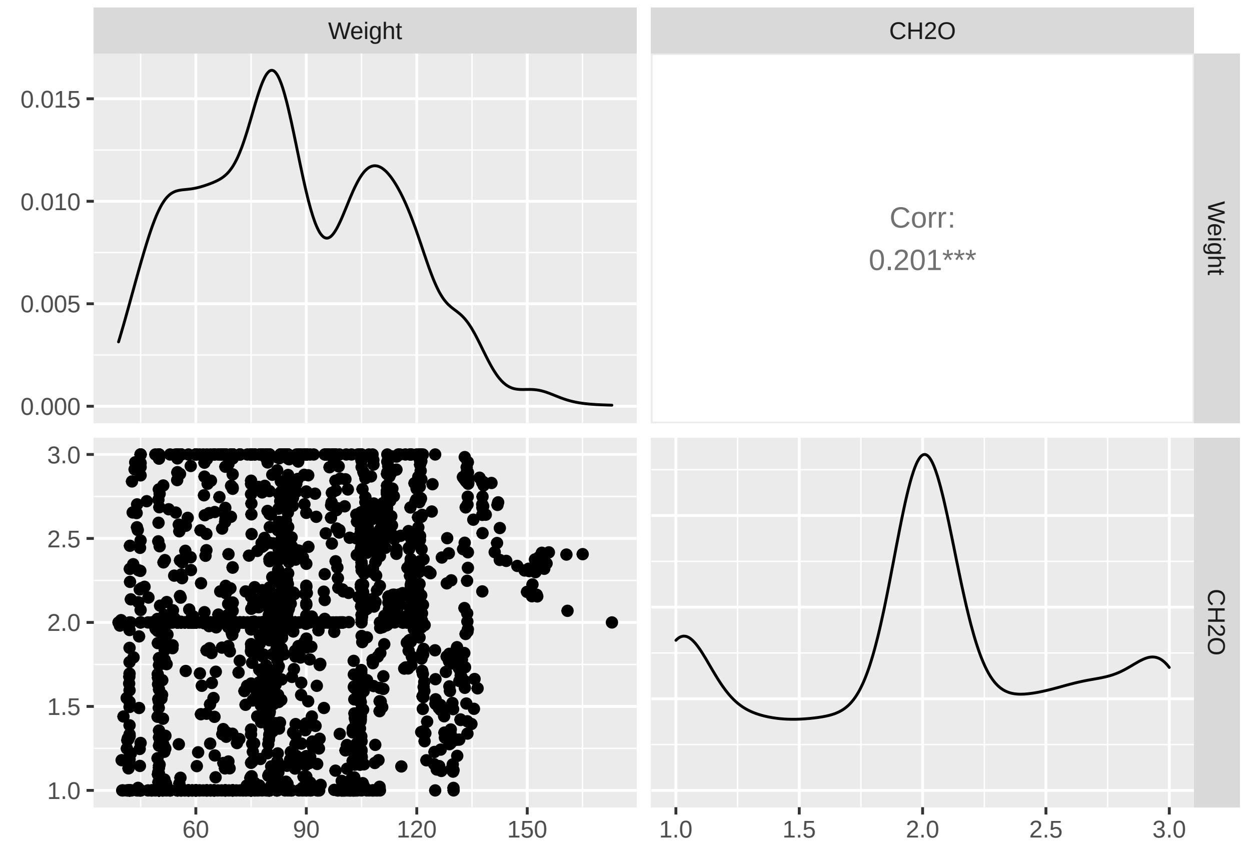
```r
# Weight vs other factors
ggpairs(obesity[c(4, 7)])
```



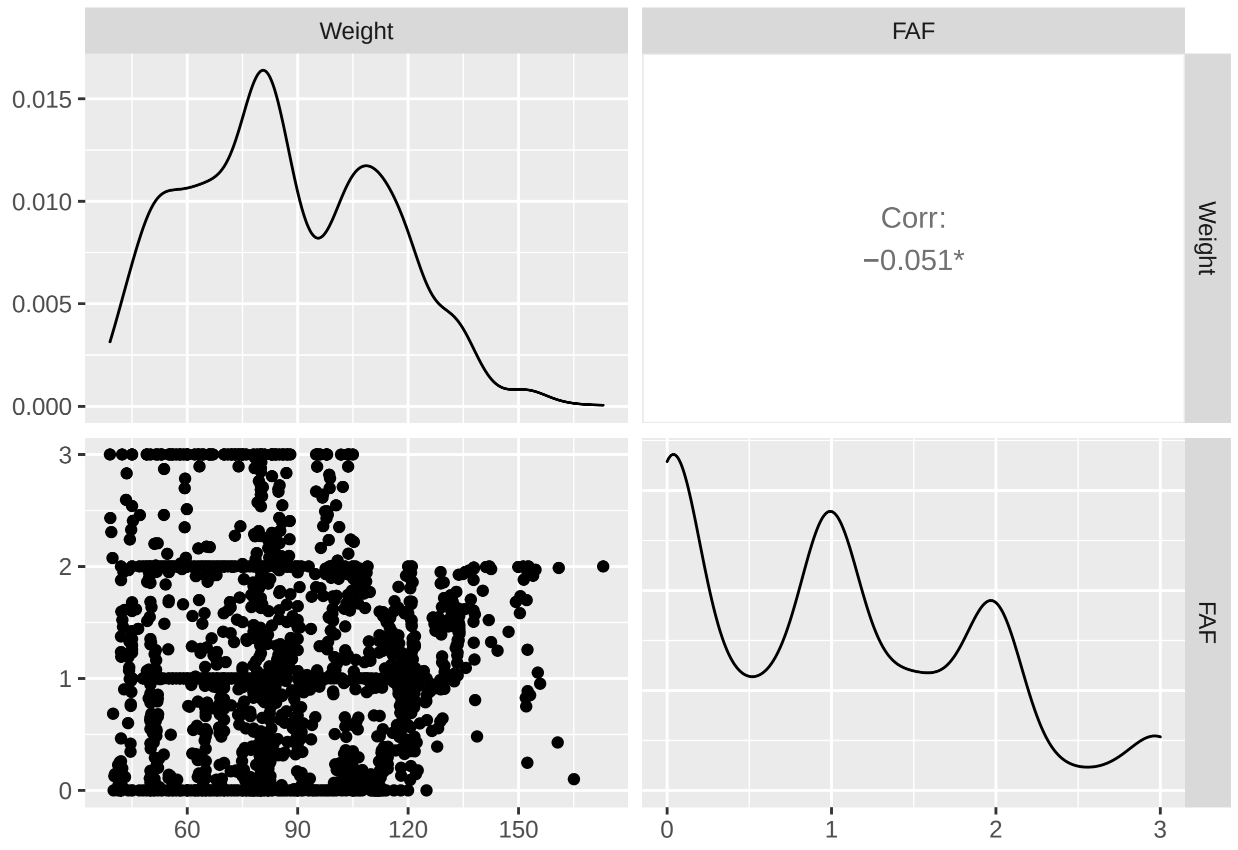
```
ggpairs(obesity[c(4, 8)])
```



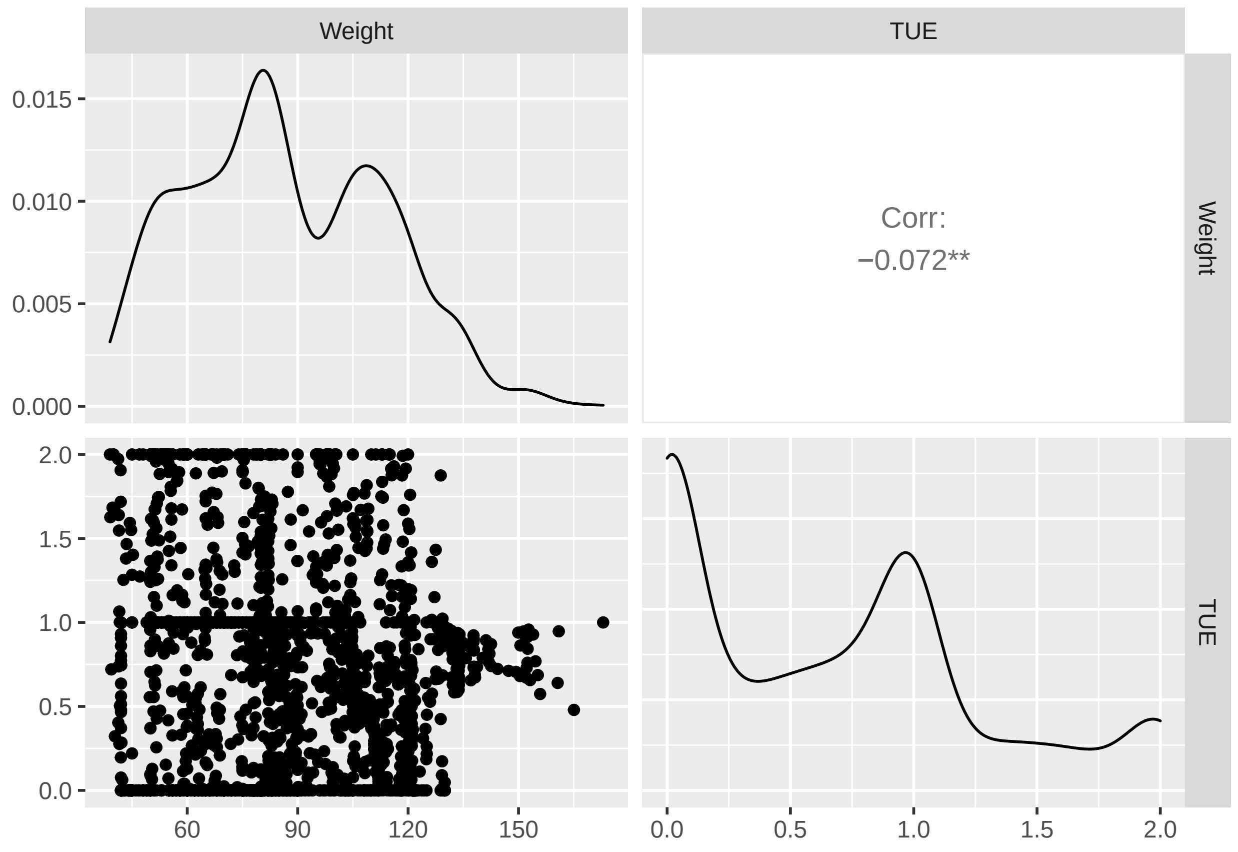
```r
ggpairs(obesity[c(4, 11)])
```



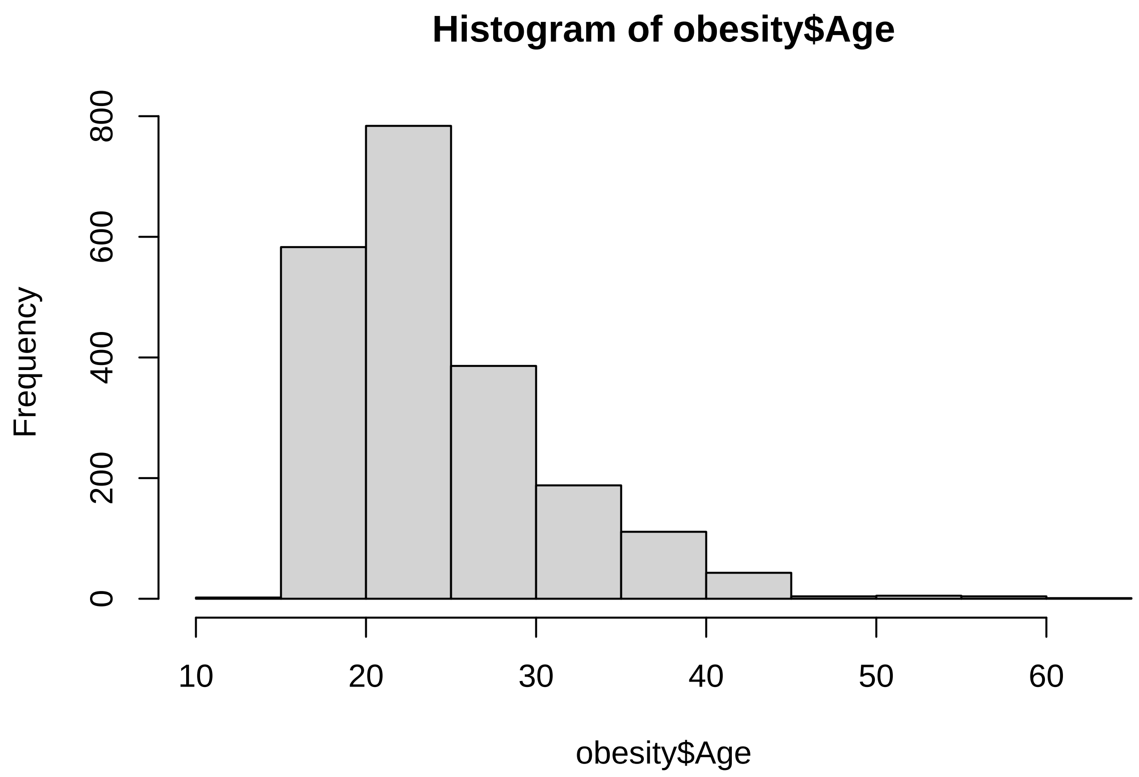
```r
ggpairs(obesity[c(4, 13)])
```



```r
ggpairs(obesity[c(4, 14)])
```

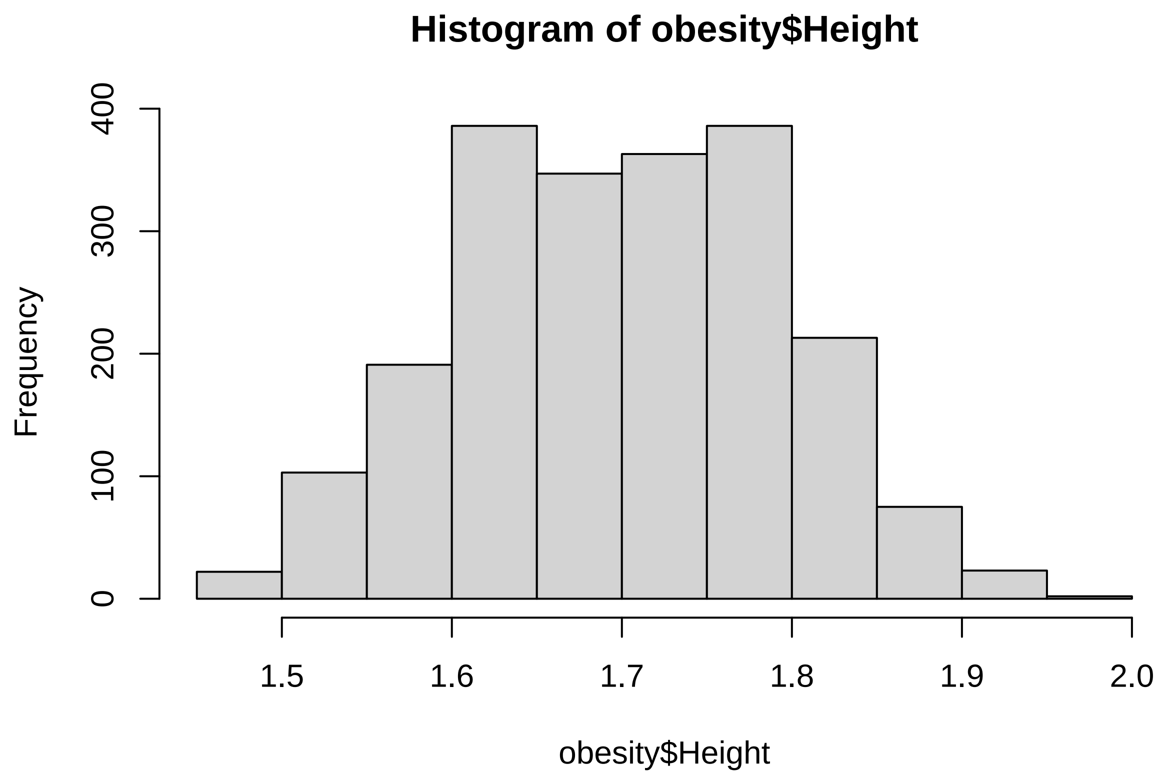


```r
# Outliers for Continuous variables
hist(obesity$Age)
```

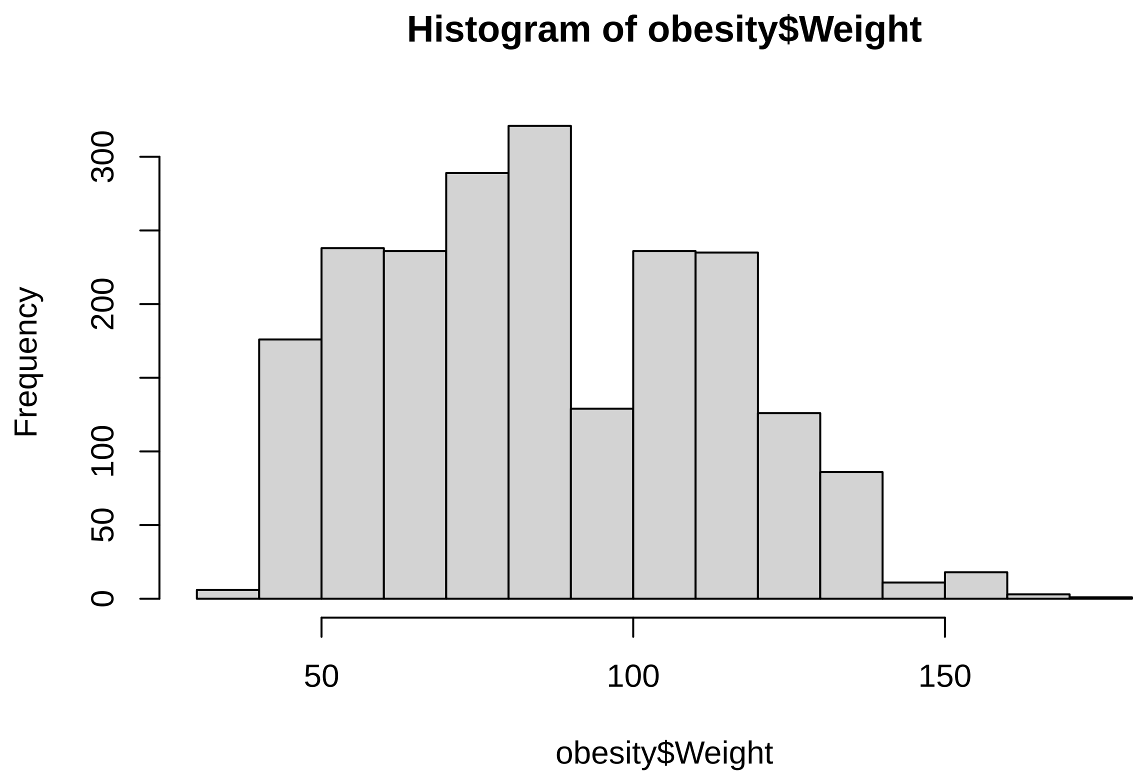


Outliers

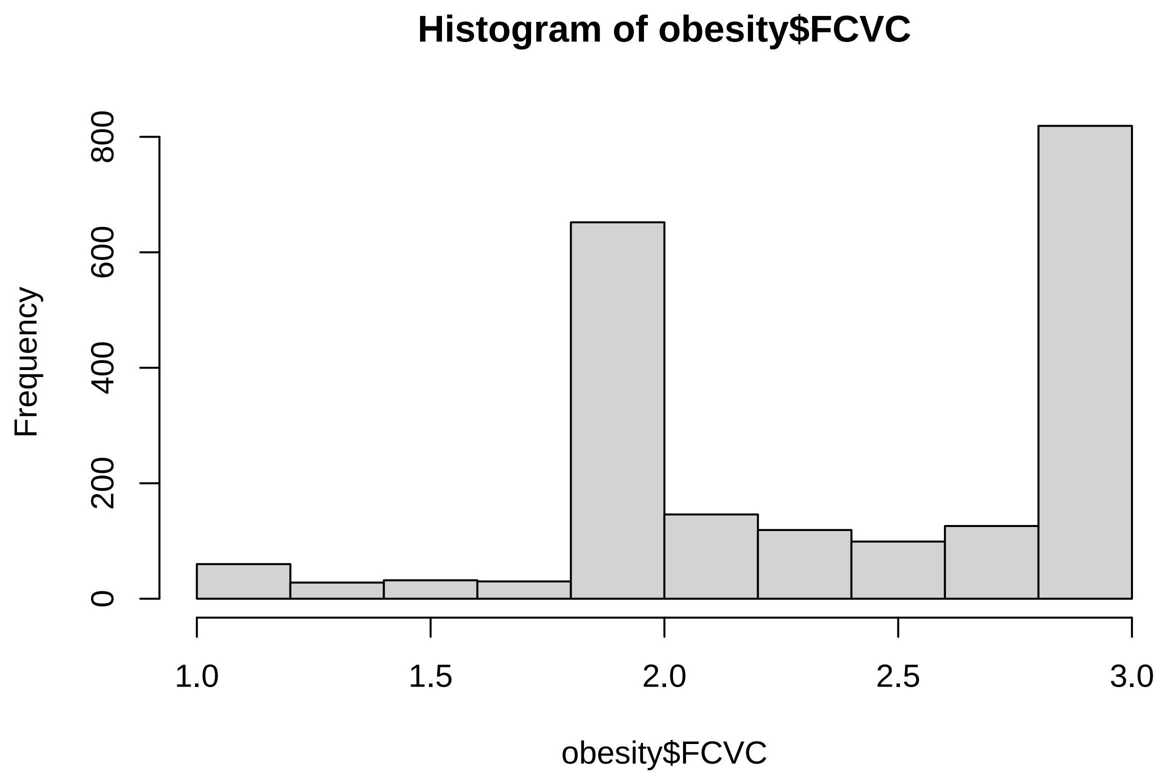
```r
hist(obesity$Height)
```



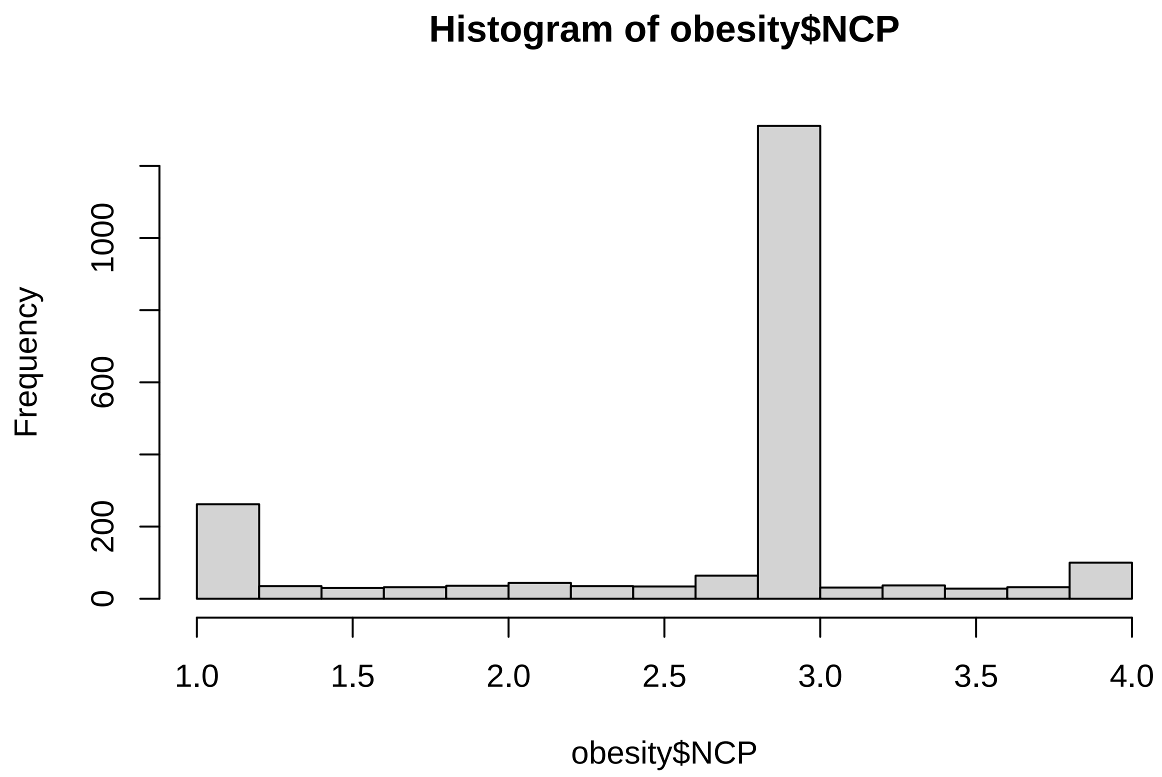
```
hist(obesity$Weight)
```



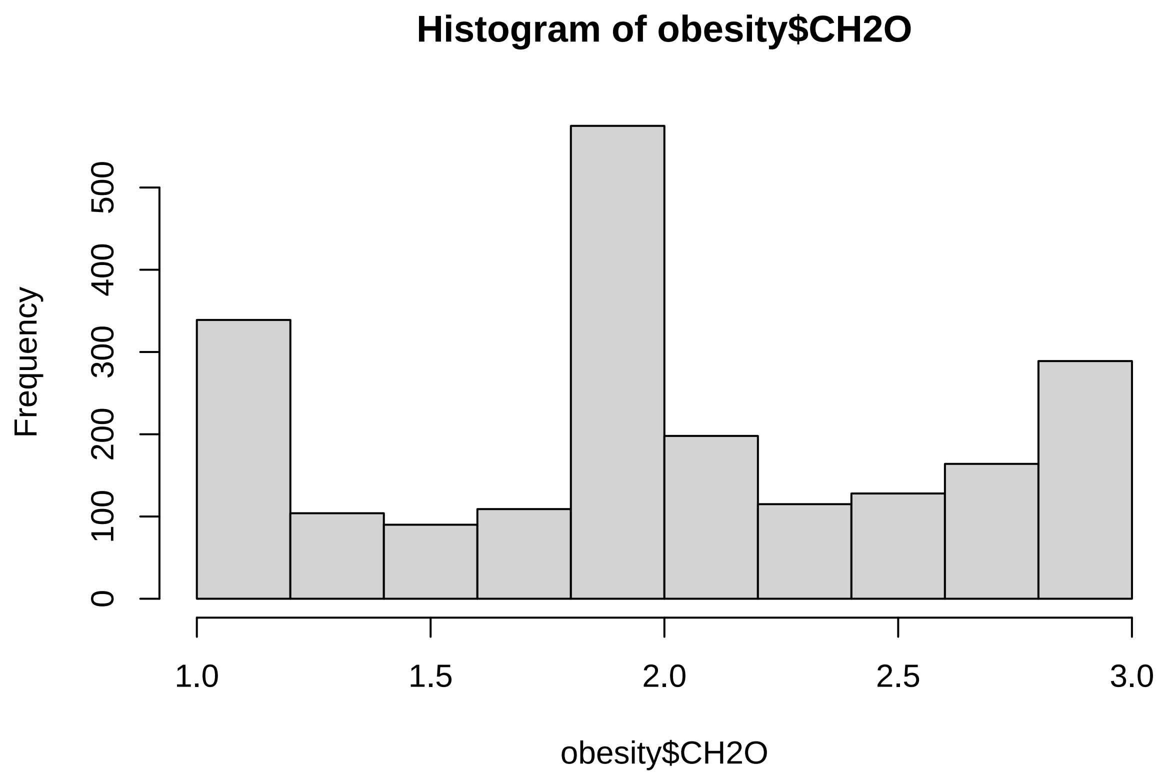
```r
hist(obesity$FCVC)
```



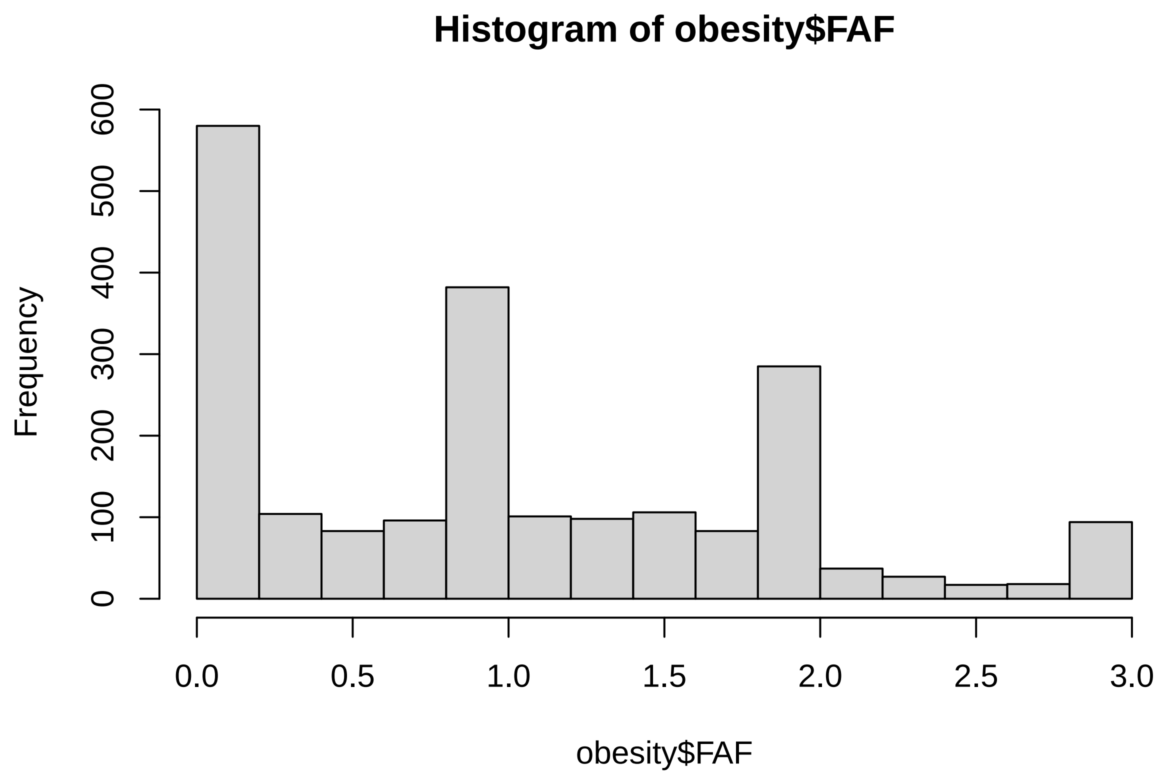
```r
hist(obesity$NCP)
```



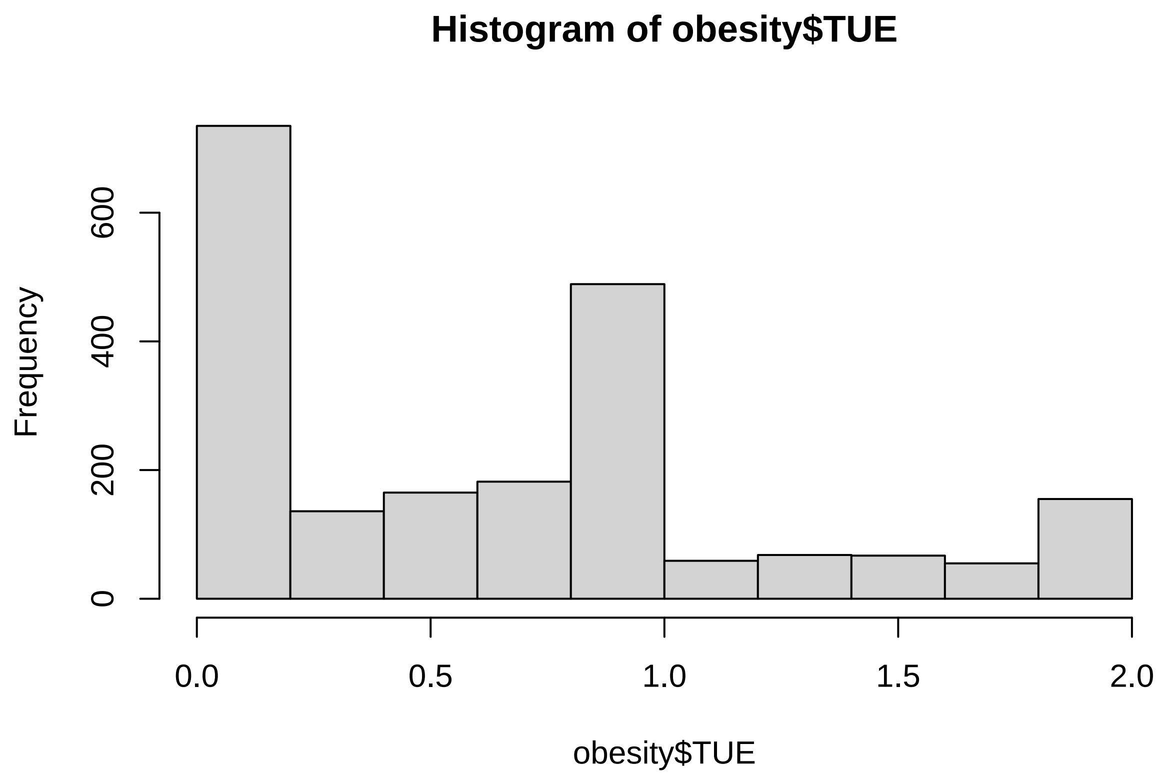
```r
hist(obesity$CH2O)
```



```r
hist(obesity$FAF)
```



```r
hist(obesity$TUE)
```



3. Data preparation - How do we organize the data for modeling?

Depending on the model, I have to convert some of the categorical variables to numerical. Using the original dataset, I duplicated three extra copies for each predictive model.

Duplicate Datasets

```
NB <- obesity
kNN <- obesity
Reg <- obesity
```

Naive Bayes Categorical Bins

To create bins of equal interval size, I used the cut() function, and called for 3 breaks with a label vector with names. That way, I didn't need to mathematically calculate the intervals themselves.

```
# Age
summary(NB$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.00   19.95   22.78   24.31   26.00   61.00
```

```r
NB$Age <- cut(NB$Age, breaks = 3, labels = c("Young", "Middle-Aged", "Old"))

# Height
summary(NB$Height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.450   1.630   1.700   1.702   1.768   1.980
```

```r
NB$Height <- cut(NB$Height, breaks = 3, labels = c("Short", "Normal", "Tall"))

# Weight
summary(NB$Weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   39.00   65.47   83.00   86.59  107.43  173.00
```

```r
NB$Weight <- cut(NB$Weight, breaks = 3, labels = c("Underweight", "Normal",
    "Overweight"))

# FCVC
summary(NB$FCVC)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.386   2.419   3.000   3.000
```

```r
NB$FCVC <- cut(NB$FCVC, breaks = 3, labels = c("Low", "Medium", "High"))

# NCP
summary(NB$NCP)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.659   3.000   2.686   3.000   4.000
```

```r
NB$NCP <- cut(NB$NCP, breaks = 3, labels = c("Low", "Medium", "High"))

# CH2O
summary(NB$CH2O)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.585   2.000   2.008   2.477   3.000
```

```r
NB$CH2O <- cut(NB$CH2O, breaks = 3, labels = c("Low", "Medium", "High"))

# FAF
summary(NB$FAF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1245  1.0000  1.0103  1.6667  3.0000
```

```r
NB$FAF <- cut(NB$FAF, breaks = 3, labels = c("Low", "Medium", "High"))

# TUE
summary(NB$TUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.6253  0.6579  1.0000  2.0000
```

```r
NB$TUE <- cut(NB$TUE, breaks = 3, labels = c("Low", "Normal", "Excessive"))
```

k Nearest Neighbors (kNN) Normalization

For kNN, we must normalize the numerical data.

```r
# Normalization Function
normalize <- function(x) {
    return((x - min(x))/(max(x) - min(x)))
}

# Get normalized data
norm_df <- as.data.frame(lapply(kNN[c(2:4, 7:8, 11, 13:14)], normalize))

# Remove Initial Columns
kNN <- subset(kNN, select = -c(2:4, 7:8, 11, 13:14))

# Replace with normalized data
kNN <- cbind(kNN, norm_df)

# Remove norm_df for memory's sake
remove(norm_df)
```

kNN Categorical to Numerical

For some of the categorical variables, we can simply convert from no/yes to 0/1.

```r
# Gender
unique(kNN$Gender)
```

```
## [1] "Female" "Male"
```

```r
kNN$Gender <- ifelse(kNN$Gender == "Male", 1, 0)

# Family History - Overweight
unique(kNN$family_history_with_overweight)
```

```
## [1] "yes" "no"
```

```r
kNN$family_history_with_overweight <- ifelse(kNN$family_history_with_overweight ==
    "yes", 1, 0)

# FAVC (Do you frequently consume high caloric food)
unique(kNN$FAVC)
```

```
## [1] "no"  "yes"
```

```r
kNN$FAVC <- ifelse(kNN$FAVC == "yes", 1, 0)

# SMOKE (Do you smoke?)
unique(kNN$SMOKE)
```

```
## [1] "no"  "yes"
```

```r
kNN$SMOKE <- ifelse(kNN$SMOKE == "yes", 1, 0)

# SCC (Do you monitor your calorie consumption?)
unique(kNN$SCC)
```

```
## [1] "no"  "yes"
```

```r
kNN$SCC <- ifelse(kNN$SCC == "yes", 1, 0)
```

kNN One-Hot Encoding

For categorical variables with more than 2 categories, we can one-hot encode them by creating new columns for each column value, and using 0/1.

```r
# Create dummy code
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

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

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

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

```r
dmy <- dummyVars(" ~ .", data = kNN)

# Create new data frame based off the dummy code
oneHot <- data.frame(predict(dmy, newdata = kNN))
```

```
# Remove the one-hot encoding for the obesity type (predicted variable)
oneHot <- subset(oneHot, select = -c(19:26))

# Add the original obesity type column, and rename
kNN <- cbind(oneHot, kNN$NObeyesdad)
names(kNN)[26] <- "NObeyesdad"

# Remove oneHot and dmy from memory
remove(dmy, oneHot)
```

Regression

```
# Use kNN dataset for regression
Reg <- kNN

# Convert obesity types to numeric values
unique(Reg$NObeyesdad)
```

```
## [1] "Normal_Weight"       "Overweight_Level_I"  "Overweight_Level_II"
## [4] "Obesity_Type_I"      "Insufficient_Weight" "Obesity_Type_II"
## [7] "Obesity_Type_III"
```

```
Reg$NObeyesdad <- factor(Reg$NObeyesdad)
Reg$NObeyesdad <- as.numeric(Reg$NObeyesdad)
unique(Reg$NObeyesdad)
```

```
## [1] 2 6 7 3 1 4 5
```

4. Modeling - What modeling techniques should we apply?

The three models I used were Naive Bayes, kNN, and Regression. I then created an ensemble of the three models that I will describe in further detail later on.

Naive Bayes

```
# Random seed
set.seed(1234)

# Training/Validation
sample <- sample.int(n = nrow(NB), size = floor(0.75 * nrow(NB)), replace = F)
NB_training <- NB[sample, ]
NB_validation <- NB[-sample, ]

# Naive Bayes Model and Prediction
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.3
```

```r
NB_model <- naiveBayes(as.factor(NObeyesdad) ~ ., data = NB_training)
NB_pred <- predict(NB_model, NB_validation)

# Results
table(Prediction = NB_pred, Actual = NB_validation$NObeyesdad)
```

```
##                      Actual
## Prediction            Insufficient_Weight Normal_Weight Obesity_Type_I
##   Insufficient_Weight                  42            18              6
##   Normal_Weight                         6            32              3
##   Obesity_Type_I                        1             1             47
##   Obesity_Type_II                       0             0             21
##   Obesity_Type_III                      0             0              0
##   Overweight_Level_I                    4            11             10
##   Overweight_Level_II                   3            11              4
##                      Actual
## Prediction            Obesity_Type_II Obesity_Type_III Overweight_Level_I
##   Insufficient_Weight               0                0                  3
##   Normal_Weight                     0                0                  9
##   Obesity_Type_I                   12                0                  8
##   Obesity_Type_II                  71                0                  9
##   Obesity_Type_III                  0               81                  0
##   Overweight_Level_I                1                0                 26
##   Overweight_Level_II               3                0                 15
##                      Actual
## Prediction            Overweight_Level_II
##   Insufficient_Weight                   4
##   Normal_Weight                        10
##   Obesity_Type_I                        9
##   Obesity_Type_II                      17
##   Obesity_Type_III                      0
##   Overweight_Level_I                    4
##   Overweight_Level_II                  26
```

kNN

I used k=10 as the standard. This yielded a very successful model that I elaborated on later in the report.

```r
# Training/Validation Data
kNN_train <- kNN[sample, ]
kNN_validation <- kNN[-sample, ]

# Test/Validation Labels
kNN_train_labels <- kNN_train[, 26]
kNN_validation_labels <- kNN_validation[, 26]

# Remove Labels from datasets
kNN_train <- subset(kNN_train, select = -26)
kNN_validation <- subset(kNN_validation, select = -26)

# Model
library(class)
```

```
## Warning: package 'class' was built under R version 4.0.3
```

```
kNN_test_pred <- knn(train = kNN_train, test = kNN_validation, cl = kNN_train_labels,
    k = 10)
kNN_test_pred <- as.vector(kNN_test_pred)

# Results
table(Prediction = kNN_test_pred, Actual = kNN_validation_labels)
```

```
##                       Actual
## Prediction             Insufficient_Weight Normal_Weight Obesity_Type_I
##   Insufficient_Weight                   50            15              2
##   Normal_Weight                          4            25              2
##   Obesity_Type_I                         1             7             72
##   Obesity_Type_II                        0             0              4
##   Obesity_Type_III                       0             1              1
##   Overweight_Level_I                     0             6              3
##   Overweight_Level_II                    1            19              7
##                       Actual
## Prediction             Obesity_Type_II Obesity_Type_III Overweight_Level_I
##   Insufficient_Weight                2                0                  6
##   Normal_Weight                      0                0                  4
##   Obesity_Type_I                     5                0                 10
##   Obesity_Type_II                   79                0                  4
##   Obesity_Type_III                   1               81                  0
##   Overweight_Level_I                 0                0                 44
##   Overweight_Level_II                0                0                  2
##                       Actual
## Prediction             Overweight_Level_II
##   Insufficient_Weight                    4
##   Normal_Weight                          5
##   Obesity_Type_I                         9
##   Obesity_Type_II                        6
##   Obesity_Type_III                       1
##   Overweight_Level_I                     8
##   Overweight_Level_II                   37
```

Regression

Since I included all factors in the Naive Bayes and kNN models, I figured it would only be appropriate to include all variables in the regression model.

```
# Training/Validation Data
reg_train <- Reg[sample, ]
reg_validation <- Reg[-sample, ]

reg_model <- glm(NObeyesdad ~ ., data = reg_train)
reg_pred <- predict(reg_model, reg_validation, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```r
# Round values
table(Prediction = round(reg_pred), Actual = reg_validation$NObeyesdad)
```

```
##           Actual
## Prediction  1  2  3  4  5  6  7
##          1  7  4  0  0  0  1  0
##          2 22 18  1  0  0  0  1
##          3 26 25  2  1  0 14  3
##          4  1 19 44  9  0 43 45
##          5  0  6 43 75 80 12 20
##          6  0  1  1  0  1  0  1
##          7  0  0  0  2  0  0  0
```

Ensemble

Rather than just looking at an individual case, this ensemble looks at the validation dataset, and gets all the predictions where the most popular prediction is the final prediction per case. The models are already created, so I felt it was redundant to run another model. If you have an individual case, you can add it to the validation dataset, where you can then run it against all 3 models (Naive Bayes, kNN, and Regression).

```r
ensemble <- function(obesity) {
    obesity_ensemble <- obesity
    obesity_ensemble$NB <- as.vector(NB_pred)
    obesity_ensemble$kNN <- kNN_test_pred
    obesity_ensemble$Reg <- round(reg_pred)

    # Substitute Numeric Values for Regression
    obesity_ensemble$Reg[obesity_ensemble$Reg == 1] <- "Insufficient_Weight"
    obesity_ensemble$Reg[obesity_ensemble$Reg == 2] <- "Normal_Weight"
    obesity_ensemble$Reg[obesity_ensemble$Reg == 3] <- "Obesity_Type_I"
    obesity_ensemble$Reg[obesity_ensemble$Reg == 4] <- "Obesity_Type_II"
    obesity_ensemble$Reg[obesity_ensemble$Reg == 5] <- "Obesity_Type_III"
    obesity_ensemble$Reg[obesity_ensemble$Reg == 6] <- "Overweight_Level_I"
    obesity_ensemble$Reg[obesity_ensemble$Reg == 7] <- "Overweight_Level_II"

    for (i in 1:length(obesity_ensemble$NB)) {
        pred_vec <- rep(NA, 3)
        pred_vec[1] <- obesity_ensemble$NB[i]
        pred_vec[2] <- obesity_ensemble$kNN[i]
        pred_vec[3] <- obesity_ensemble$Reg[i]

        if (pred_vec[1] == pred_vec[2] | pred_vec[2] == pred_vec[3]) {
            obesity_ensemble$best_pred[i] <- pred_vec[2]
        } else if (pred_vec[1] == pred_vec[3] | pred_vec[2] == pred_vec[3]) {
            obesity_ensemble$best_pred[i] <- pred_vec[3]
        } else {
            # Get Numerical Values, and get the medium value
            replace(pred_vec, pred_vec == "Insufficient_Weight", 1)
            replace(pred_vec, pred_vec == "Normal_Weight", 2)
            replace(pred_vec, pred_vec == "Obesity_Type_I", 3)
            replace(pred_vec, pred_vec == "Obesity_Type_II", 4)
            replace(pred_vec, pred_vec == "Obesity_Type_III", 5)
            replace(pred_vec, pred_vec == "Overweight_Level_I", 6)
            replace(pred_vec, pred_vec == "Overweight_Level_II", 7)

            new_val <- median(pred_vec)

            # Replace back the numerical value with the actual prediction
            replace(new_val, new_val == 1, "Insufficient_Weight")
            replace(new_val, new_val == 2, "Normal_Weight")
            replace(new_val, new_val == 3, "Obesity_Type_I")
            replace(new_val, new_val == 4, "Obesity_Type_II")
            replace(new_val, new_val == 5, "Obesity_Type_III")
            replace(new_val, new_val == 6, "Overweight_Level_I")
            replace(new_val, new_val == 7, "Overweight_Level_II")

            obesity_ensemble$best_pred[i] <- new_val
        }

        # Overall Prediction
        outcome <- confusionMatrix(as.factor(obesity_ensemble$best_pred), as.factor(obesity_ensemble$NO
        return(outcome)
    }
}
```

```
ensemble(NB_validation)
```

```
## Confusion Matrix and Statistics
##
##                      Reference
## Prediction            Insufficient_Weight Normal_Weight Obesity_Type_I
##   Insufficient_Weight                   0             0              0
##   Normal_Weight                         0             0              0
##   Obesity_Type_I                        0             0              0
##   Obesity_Type_II                       0             0              0
##   Obesity_Type_III                     56            73             91
##   Overweight_Level_I                    0             0              0
##   Overweight_Level_II                   0             0              0
##                      Reference
## Prediction            Obesity_Type_II Obesity_Type_III Overweight_Level_I
##   Insufficient_Weight               0                0                  0
##   Normal_Weight                     0                0                  0
##   Obesity_Type_I                    0                0                  0
##   Obesity_Type_II                   0                0                  0
##   Obesity_Type_III                 87               81                 70
##   Overweight_Level_I                0                0                  0
##   Overweight_Level_II               0                0                  0
##                      Reference
## Prediction            Overweight_Level_II
##   Insufficient_Weight                   0
##   Normal_Weight                         0
##   Obesity_Type_I                        0
##   Obesity_Type_II                       0
##   Obesity_Type_III                     70
##   Overweight_Level_I                    0
##   Overweight_Level_II                   0
##
## Overall Statistics
##
##                Accuracy : 0.1534
##                  95% CI : (0.1237, 0.187)
##     No Information Rate : 0.1723
##     P-Value [Acc > NIR] : 0.8881
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Insufficient_Weight Class: Normal_Weight
## Sensitivity                              0.0000               0.0000
## Specificity                              1.0000               1.0000
## Pos Pred Value                              NaN                  NaN
## Neg Pred Value                           0.8939               0.8617
## Prevalence                               0.1061               0.1383
## Detection Rate                           0.0000               0.0000
## Detection Prevalence                     0.0000               0.0000
## Balanced Accuracy                        0.5000               0.5000
##                      Class: Obesity_Type_I Class: Obesity_Type_II
## Sensitivity                         0.0000                 0.0000
## Specificity                         1.0000                 1.0000
## Pos Pred Value                         NaN                    NaN
## Neg Pred Value                      0.8277                 0.8352
## Prevalence                          0.1723                 0.1648
## Detection Rate                      0.0000                 0.0000
## Detection Prevalence                0.0000                 0.0000
## Balanced Accuracy                   0.5000                 0.5000
##                      Class: Obesity_Type_III Class: Overweight_Level_I
## Sensitivity                           1.0000                    0.0000
## Specificity                           0.0000                    1.0000
## Pos Pred Value                        0.1534                       NaN
## Neg Pred Value                           NaN                    0.8674
## Prevalence                            0.1534                    0.1326
## Detection Rate                        0.1534                    0.0000
## Detection Prevalence                  1.0000                    0.0000
## Balanced Accuracy                     0.5000                    0.5000
##                      Class: Overweight_Level_II
## Sensitivity                              0.0000
## Specificity                              1.0000
## Pos Pred Value                              NaN
## Neg Pred Value                           0.8674
## Prevalence                               0.1326
## Detection Rate                           0.0000
## Detection Prevalence                     0.0000
## Balanced Accuracy                        0.5000
```

I have a feeling this ensemble did not have a good accuracy/recall/specificity because it put the regression model into consideration. This should not have been the case because the regression model was not reliable (low accuracy, precision, etc). That is something I learned from creating this ensemble. For the future, Naive Bayes and kNN look like safer bets for non-binary classification problems.

5. Evaluation - Which model best meets the business objectives?

For classification problems, we can look at accuracy, precision, recall/sensitivity, and specificity. Here is an overview of each metric. Accuracy, recall/sensitivity, and specificity are already in the confusion matrix results. Due to this problem not being binary, I felt that it would have been repetitive and redundant to include precision since it the confusion matrix does not include that statistic.

1. Accuracy is number of correct predictions made over the total number of predictions made.
2. Precision looks at all of the true positives predictions compared to all of the positive predictions
3. Recall/Sensitivity looks at all of the true positive predictions compared to all of the actual positive results.
4. Specificity looks at all of the true negative predictions compared to all of the actual negative results.

Naive Bayes Results

The accuracy was mediocre, but not terrible. Apart from the accuracy, the sensitivity and specificity was decent for certain classes, while it was poor for other classes. For normal weight and overweight level 2, the sensitivity was very poor, suggesting that there was too many predictions for those categories, even when the actual category was not the mentioned category.

```
# Confusion Matrix
confusionMatrix(as.factor(NB_pred), as.factor(NB_validation$NObeyesdad))
```

```
## Confusion Matrix and Statistics
##
##                      Reference
## Prediction            Insufficient_Weight Normal_Weight Obesity_Type_I
##   Insufficient_Weight                  42            18              6
##   Normal_Weight                         6            32              3
##   Obesity_Type_I                        1             1             47
##   Obesity_Type_II                       0             0             21
##   Obesity_Type_III                      0             0              0
##   Overweight_Level_I                    4            11             10
##   Overweight_Level_II                   3            11              4
##                      Reference
## Prediction            Obesity_Type_II Obesity_Type_III Overweight_Level_I
##   Insufficient_Weight               0                0                  3
##   Normal_Weight                     0                0                  9
##   Obesity_Type_I                   12                0                  8
##   Obesity_Type_II                  71                0                  9
##   Obesity_Type_III                  0               81                  0
##   Overweight_Level_I                1                0                 26
##   Overweight_Level_II               3                0                 15
##                      Reference
## Prediction            Overweight_Level_II
##   Insufficient_Weight                   4
##   Normal_Weight                        10
##   Obesity_Type_I                        9
##   Obesity_Type_II                      17
##   Obesity_Type_III                    0
##   Overweight_Level_I                  4
##   Overweight_Level_II                26
##
## Overall Statistics
##
##                Accuracy : 0.6155
##                  95% CI : (0.5725, 0.6572)
##     No Information Rate : 0.1723
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5499
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Insufficient_Weight Class: Normal_Weight
## Sensitivity                             0.75000              0.43836
## Specificity                             0.93432              0.93846
## Pos Pred Value                          0.57534              0.53333
## Neg Pred Value                          0.96923              0.91239
## Prevalence                              0.10606              0.13826
## Detection Rate                          0.07955              0.06061
## Detection Prevalence                    0.13826              0.11364
## Balanced Accuracy                       0.84216              0.68841
##                      Class: Obesity_Type_I Class: Obesity_Type_II
## Sensitivity                        0.51648                 0.8161
## Specificity                        0.92906                 0.8934
## Pos Pred Value                     0.60256                 0.6017
## Neg Pred Value                     0.90222                 0.9610
## Prevalence                         0.17235                 0.1648
## Detection Rate                     0.08902                 0.1345
## Detection Prevalence               0.14773                 0.2235
## Balanced Accuracy                  0.72277                 0.8548
##                      Class: Obesity_Type_III Class: Overweight_Level_I
## Sensitivity                           1.0000                   0.37143
## Specificity                           1.0000                   0.93450
## Pos Pred Value                        1.0000                   0.46429
## Neg Pred Value                        1.0000                   0.90678
## Prevalence                            0.1534                   0.13258
## Detection Rate                        0.1534                   0.04924
## Detection Prevalence                  0.1534                   0.10606
## Balanced Accuracy                     1.0000                   0.65296
##                      Class: Overweight_Level_II
## Sensitivity                             0.37143
## Specificity                             0.92140
## Pos Pred Value                          0.41935
## Neg Pred Value                          0.90558
## Prevalence                              0.13258
## Detection Rate                          0.04924
## Detection Prevalence                    0.11742
## Balanced Accuracy                       0.64641
```

kNN Results

This model has very good accuracy, sensitivity, and specificity values across each obesity level/type. An overall accuracy of 73% is solid, and the balanced accuracies are even better (apart from the normal weight). Based off of my instinct, this seems like the best model. However, I have to still run the regression and ensemble model.

```
# Cross Table
library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 4.0.3
```

```
CrossTable(x = kNN_test_pred, y = kNN_validation_labels, prop.chisq = FALSE,
    prop.c = FALSE, prop.r = FALSE, prop.t = FALSE)
```

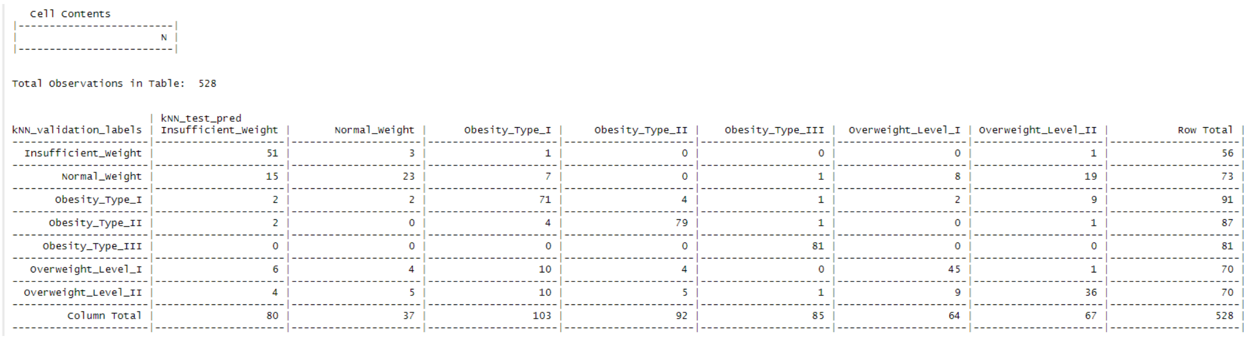
```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  528
##
##
##                      | kNN_validation_labels
##        kNN_test_pred | Insufficient_Weight |       Normal_Weight |      Obesity_Type_I |     Obesity_
## ---------------------|---------------------|---------------------|---------------------|-------------
## Insufficient_Weight |                  50 |                  15 |                   2 |
## ---------------------|---------------------|---------------------|---------------------|-------------
##        Normal_Weight |                   4 |                  25 |                   2 |
## ---------------------|---------------------|---------------------|---------------------|-------------
##       Obesity_Type_I |                   1 |                   7 |                  72 |
## ---------------------|---------------------|---------------------|---------------------|-------------
##      Obesity_Type_II |                   0 |                   0 |                   4 |
## ---------------------|---------------------|---------------------|---------------------|-------------
##     Obesity_Type_III |                   0 |                   1 |                   1 |
## ---------------------|---------------------|---------------------|---------------------|-------------
##   Overweight_Level_I |                   0 |                   6 |                   3 |
## ---------------------|---------------------|---------------------|---------------------|-------------
##  Overweight_Level_II |                   1 |                  19 |                   7 |
## ---------------------|---------------------|---------------------|---------------------|-------------
##         Column Total |                  56 |                  73 |                  91 |
## ---------------------|---------------------|---------------------|---------------------|-------------
##
##
```

```
# Confusion Matrix
confusionMatrix(as.factor(kNN_test_pred), as.factor(kNN_validation_labels))
```

```
## Confusion Matrix and Statistics
```

```
##
##                      Reference
## Prediction            Insufficient_Weight Normal_Weight Obesity_Type_I
##   Insufficient_Weight                  50            15              2
##   Normal_Weight                         4            25              2
##   Obesity_Type_I                        1             7             72
##   Obesity_Type_II                       0             0              4
##   Obesity_Type_III                      0             1              1
##   Overweight_Level_I                    0             6              3
##   Overweight_Level_II                   1            19              7
##                      Reference
## Prediction            Obesity_Type_II Obesity_Type_III Overweight_Level_I
##   Insufficient_Weight               2                0                  6
##   Normal_Weight                     0                0                  4
##   Obesity_Type_I                    5                0                 10
##   Obesity_Type_II                  79                0                  4
##   Obesity_Type_III                  1               81                  0
##   Overweight_Level_I                0                0                 44
##   Overweight_Level_II               0                0                  2
##                      Reference
## Prediction            Overweight_Level_II
##   Insufficient_Weight                   4
##   Normal_Weight                         5
##   Obesity_Type_I                        9
##   Obesity_Type_II                       6
##   Obesity_Type_III                      1
##   Overweight_Level_I                    8
##   Overweight_Level_II                  37
##
## Overall Statistics
##
##                Accuracy : 0.7348
##                  95% CI : (0.695, 0.772)
##     No Information Rate : 0.1723
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6896
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Insufficient_Weight Class: Normal_Weight
## Sensitivity                              0.8929              0.34247
## Specificity                              0.9386              0.96703
## Pos Pred Value                           0.6329              0.62500
## Neg Pred Value                           0.9866              0.90164
## Prevalence                               0.1061              0.13826
## Detection Rate                           0.0947              0.04735
## Detection Prevalence                     0.1496              0.07576
## Balanced Accuracy                        0.9157              0.65475
##                      Class: Obesity_Type_I Class: Obesity_Type_II
## Sensitivity                         0.7912                 0.9080
## Specificity                         0.9268                 0.9683
## Pos Pred Value                      0.6923                 0.8495
## Neg Pred Value                      0.9552                 0.9816
## Prevalence                          0.1723                 0.1648
## Detection Rate                      0.1364                 0.1496
## Detection Prevalence                0.1970                 0.1761
## Balanced Accuracy                   0.8590                 0.9381
##                        Class: Obesity_Type_III Class: Overweight_Level_I
## Sensitivity                           1.0000                 0.62857
## Specificity                           0.9911                 0.96288
## Pos Pred Value                        0.9529                 0.72131
## Neg Pred Value                        1.0000                 0.94433
## Prevalence                            0.1534                 0.13258
## Detection Rate                        0.1534                 0.08333
## Detection Prevalence                  0.1610                 0.11553
## Balanced Accuracy                     0.9955                 0.79573
##                        Class: Overweight_Level_II
## Sensitivity                              0.52857
## Specificity                              0.93668
## Pos Pred Value                           0.56061
## Neg Pred Value                           0.92857
## Prevalence                               0.13258
## Detection Rate                           0.07008
## Detection Prevalence                     0.12500
## Balanced Accuracy                        0.73263
```

I inserted an image of one of the CrossTables while coding because I could not get the CrossTable to fit on the page. I just stuck with the confusion matrices in the future for visual's sake since the confusion matrix gave us the same information.



```
   Cell Contents
|-------------------------|
|                       N |
|-------------------------|


Total Observations in Table:  528


                     | kNN_test_pred
kNN_validation_labels | Insufficient_Weight |       Normal_Weight |      Obesity_Type_I |     Obesity_Type_II |    Obesity_Type_III |  Overweight_Level_I | Overweight_Level_II |           Row Total |
---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
  Insufficient_Weight |                  51 |                   3 |                   1 |                   0 |                   0 |                   0 |                   1 |                  56 |
---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
        Normal_Weight |                  15 |                  23 |                   7 |                   0 |                   1 |                   8 |                  19 |                  73 |
---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
       Obesity_Type_I |                   2 |                   2 |                  71 |                   4 |                   1 |                   2 |                   9 |                  91 |
---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
      Obesity_Type_II |                   2 |                   0 |                   4 |                  79 |                   1 |                   0 |                   1 |                  87 |
---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
     Obesity_Type_III |                   0 |                   0 |                   0 |                   0 |                  81 |                   0 |                   0 |                  81 |
---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
   Overweight_Level_I |                   6 |                   4 |                  10 |                   4 |                   0 |                  45 |                   1 |                  70 |
---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
  Overweight_Level_II |                   4 |                   5 |                  10 |                   5 |                   1 |                   9 |                  36 |                  70 |
---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
         Column Total |                  80 |                  37 |                 103 |                  92 |                  85 |                  64 |                  67 |                 528 |
---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
```

Figure 1: Cross Table

Regression Results

This regression model was subpar. I believe that it did not go as planned because the regrsssion problem was not binary, and this is reflected in the accuracy and the sensitivity values (apart from the Obesity Type III [class:5])

```r
# Confusion Matrix
confusionMatrix(as.factor(round(reg_pred)), as.factor(reg_validation$NObeyesdad))
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction  1  2  3  4  5  6  7
##          1  7  4  0  0  0  1  0
##          2 22 18  1  0  0  0  1
##          3 26 25  2  1  0 14  3
##          4  1 19 44  9  0 43 45
##          5  0  6 43 75 80 12 20
##          6  0  1  1  0  1  0  1
##          7  0  0  0  2  0  0  0
##
## Overall Statistics
##
##                Accuracy : 0.2197
##                  95% CI : (0.1851, 0.2575)
##     No Information Rate : 0.1723
##     P-Value [Acc > NIR] : 0.003021
##
##                   Kappa : 0.0745
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6
## Sensitivity           0.12500  0.24658 0.021978  0.10345   0.9877 0.000000
## Specificity           0.98941  0.94725 0.842105  0.65533   0.6510 0.991266
## Pos Pred Value        0.58333  0.42857 0.028169  0.05590   0.3390 0.000000
## Neg Pred Value        0.90504  0.88683 0.805252  0.78747   0.9966 0.866412
## Prevalence            0.10606  0.13826 0.172348  0.16477   0.1534 0.132576
## Detection Rate        0.01326  0.03409 0.003788  0.01705   0.1515 0.000000
## Detection Prevalence  0.02273  0.07955 0.134470  0.30492   0.4470 0.007576
## Balanced Accuracy     0.55720  0.59691 0.432042  0.37939   0.8193 0.495633
##                      Class: 7
## Sensitivity          0.000000
## Specificity          0.995633
## Pos Pred Value       0.000000
## Neg Pred Value       0.866920
## Prevalence           0.132576
## Detection Rate       0.000000
## Detection Prevalence 0.003788
## Balanced Accuracy    0.497817
```

6. Deployment - How do stakeholders access the results?

Stakeholders merely want the results of the models, and a recommendation as to which model to use. From my analysis, it is clear that the kNN model is the most accurate predictive model for this particular dataset. It has the highest accuracy value, the highest sensitivity values, and the highest specificity values across the board. This makes kNN the obvious choice. This doesn't surprise me because kNN shows little to no bias in it's models, which is helpful for classification models with a lot of variables/outcomes (non-binary classification with 17 variables).