

Data Mining Project (MaBAn 2020)

Predicting obesity levels according to daily habits

by : Ángel Tomás-Ripoll & Laurence Tétreault-Falsafi

Contents

Introduction	1
Data Pre-Processing	5
Exploratory Data Analysis	13
Data Analysis	23
Multiple Linear Regression	23
k-Nearest Neighbors	46
Regression Tree	50
Ensemble Method (MLR + k-NN + Regression Tree)	64
Creating a person for prediction	64

Introduction

For this project, our objective is to predict the expected weight level (in Kg) for a given person depending on certain daily habits (eating and physical activity) and on the person's age, gender and height.

To do this, we found a quite interesting dataset (click here : <http://archive.ics.uci.edu/ml/datasets/Estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition+>) containing 2111 observations and 17 variables (mainly categorical).

Please, find here a manually created metadata table :

```
# To adjust the page margins when knitting to PDF :

library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=45),tidy=TRUE)
```

```
# Used packages :
```

```
library(pander)
library(dplyr)
library(gt)
library(car)
library(ggplot2)
library(gridExtra)
library(psych)
library(corrplot)
library(ellipse)
library(dummies)
library(nnet)
library(class)
library(caret)
library(rpart)
library(rpart.plot)
library(ehaGoF)
library(forecast)
library(randomForest)
```

```
# Working Directory :
```

```
setwd("~/GitHub/CVTDM_Project_MaBAn_2020")
```

```
# Reading the data :
```

```
obesity <- read.csv("Obesity.csv", header = T,
  sep = ",")
attach(obesity)
```

```
obesity_original <- obesity
```

```
# Small metadata table :
```

```
tibble_table <- tibble(`Variable Name` = c(colnames(obesity)[1:14],
  "", colnames(obesity)[15:17]), Description = c("Gender",
  "Age", "Height", "Weight", "Has a family member suffered or suffers from overweight?",
  "Do you eat high caloric food frequently?",
  "Do you usually eat vegetables in your meals?",
  "How many main meals do you have daily?",
  "Do you eat any food between meals?", "Do you smoke?",
  "How much water do you drink daily?", "Do you monitor the calories you eat daily?",
  "How often do you have physical activity?",
```

```

    "How much time do you use technological devices such as",
    "cell phone videogames, television, computer and others?",
    "How often do you drink alcohol?", "Which transportation do you usually use?",
    "Obesity level based on calculation of Mass Body Index"))

metadata <- gt(data = tibble_table)

metadata %>% tab_header(title = md("**Metadata**"),
  subtitle = "from the dataset we are using") %>%

tab_source_note(source_note = "Based on information in :

https://www.sciencedirect.com/science/article/pii/S2352340919306985")

```

Metadata

from the dataset we are using

Variable Name	Description
Gender	Gender
Age	Age
Height	Height
Weight	Weight
family_history_with_overweight	Has a family member suffered or suffers from overweight?
FAVC	Do you eat high caloric food frequently?
FCVC	Do you usually eat vegetables in your meals?
NCP	How many main meals do you have daily?
CAEC	Do you eat any food between meals?
SMOKE	Do you smoke?
CH2O	How much water do you drink daily?
SCC	Do you monitor the calories you eat daily?
FAF	How often do you have physical activity?
TUE	How much time do you use technological devices such as cell phone videogames, television, computer and others?
CALC	How often do you drink alcohol?
MTRANS	Which transportation do you usually use?
NObeyesdad	Obesity level based on calculation of Mass Body Index

Based on information in :

<https://www.sciencedirect.com/science/article/pii/S2352340919306985>

Here is a small overview of the first observations :

```
pander(head(obesity))
```

Table continues below

Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC
Female	21	1.62	64	yes	no	2
Female	21	1.52	56	yes	no	3
Male	23	1.8	77	yes	no	2
Male	27	1.8	87	no	no	3
Male	22	1.78	89.8	no	no	2
Male	29	1.62	53	no	yes	2

Table continues below

NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC
3	Sometimes	no	2	no	0	1	no
3	Sometimes	yes	3	yes	3	0	Sometimes
3	Sometimes	no	2	no	2	1	Frequently
3	Sometimes	no	2	no	2	0	Frequently
1	Sometimes	no	2	no	0	0	Sometimes
3	Sometimes	no	2	no	0	0	Sometimes

MTRANS	NObeyesdad
Public_Transportation	Normal_Weight
Public_Transportation	Normal_Weight
Public_Transportation	Normal_Weight
Walking	Overweight_Level_I
Public_Transportation	Overweight_Level_II
Automobile	Normal_Weight

The variable of interest is “Weight”, it will be our dependent variable.

This data set seems to be of high quality, because it has no missing observations, and our subsequent exploratory analysis will tell us if there are outliers to be handled with.

We will first begin with a basic data pre-processing which will be followed by a Data Exploratory Analysis. We will develop several models in order to accurately predict the level of weight of each individual.

The models will be :

1. Multiple Linear Regression
2. Regression tree
3. k-Nearest Neighbors
4. Ensemble Method

We will deploy the best model based on error metrics and prediction performance.

At the very end, we will make a **Shiny App** available, in which any user can fill-in a questionnaire concerning daily habits, age and height. Then, the App will tell the user what is the expected weight according to those characteristics, and will present the result in two forms :

- The expected weight in Kg.
- The expected obesity level based on the Body Mass Index, following the classification coming from the World Health Organisation.

The user will also be able to **select the type of model** that will predict the results. That way, it will be interesting to see with just a few clicks how each model will yield different results.

Data Pre-Processing

The first thing to do is to change the column names so that they are more visually meaningful and less confusing.

```
# Changing column names:
```

```
names(obesity)[5] = "family_history"
names(obesity)[6] = "eat_caloric"
names(obesity)[7] = "vegetables"
names(obesity)[8] = "main_meals"
names(obesity)[9] = "food_inbetween"
names(obesity)[12] = "monitor_cal"
names(obesity)[13] = "physical_act"
names(obesity)[14] = "tech_devices"
names(obesity)[15] = "alcohol"
```

```
# Checking the dataset structure :
```

```
pander(str(obesity))
```

```
‘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:
chr “yes” “yes” “yes” “no” ... $ eat_caloric : chr “no” “no” “no” “no” ... $ vegetables :
```

```

num 2 3 2 3 2 2 3 2 3 2 ... $ main_meals : num 3 3 3 3 1 3 3 3 3 3 ... $ food_inbetween:
chr "Sometimes" "Sometimes" "Sometimes" "Sometimes" ... $ SMOKE : chr "no" "yes"
"no" "no" ... $ CH2O : num 2 3 2 2 2 2 2 2 2 2 ... $ monitor_cal : chr "no" "yes" "no"
"no" ... $ physical_act : num 0 3 2 2 0 0 1 3 1 1 ... $ tech_devices : num 1 0 1 0 0 0
0 0 1 1 ... $ alcohol : chr "no" "Sometimes" "Frequently" "Frequently" ... $ MTRANS :
chr "Public_Transportation" "Public_Transportation" "Public_Transportation" "Walking"
... $ NObeyesdad : chr "Normal_Weight" "Normal_Weight" "Normal_Weight" "Over-
weight_Level_I" ...

```

```
pander(summary(obesity[, 2:4]))
```

Age	Height	Weight
Min. :14.00	Min. :1.450	Min. : 39.00
1st Qu.:19.95	1st Qu.:1.630	1st Qu.: 65.47
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

Since many variables are numerical and continuous between a range (for example `vegetables`, inside the range 1 to 3), we will transform them into categorical. This is, somehow, BINNING. For this, we will follow the names given in the information file referred to earlier (<https://www.sciencedirect.com/science/article/pii/S2352340919306985>).

To make this task easier, we created a function that bins variables. This function is called “binning”

```

# Binning some numerical variables :

binning <- function(x) {

  # vegetables

  x$vegetables[x$vegetables <= 1] <- "Never"

  x$vegetables[x$vegetables > 1 & x$vegetables <=
    2] <- "Sometimes"

  x$vegetables[x$vegetables > 2 & x$vegetables <=
    3] <- "Always"

  # main_meals

  x$main_meals[x$main_meals >= 1 & x$main_meals <

```

```

    3] <- "Btw_1_&_2"

x$main_meals[x$main_meals == 3] <- "Three"

x$main_meals[x$main_meals > 3 & x$main_meals <=
  4] <- "More_than_3"

# tech_devices

x$tech_devices[x$tech_devices >= 0 & x$tech_devices <=
  0.5] <- "Zero_hours"

x$tech_devices[x$tech_devices <= 1.5] <- "One_hour"

x$tech_devices[x$tech_devices <= 2] <- "Two_hours"

# physical_act

x$physical_act[x$physical_act < 1] <- "I do not have"

x$physical_act[x$physical_act >= 1 & x$physical_act <=
  2] <- "1 or 2 days"

x$physical_act[x$physical_act >= 2 & x$physical_act <=
  4] <- "2 or 4 days"

x$physical_act[x$physical_act >= 4 & x$physical_act <=
  5] <- "4 or 5 days"

# CH2O

x$CH2O[x$CH2O <= 1] <- "Less than a liter"

x$CH2O[x$CH2O <= 2] <- "Between 1 and 2 L"

x$CH2O[x$CH2O <= 3] <- "More than 2 L"

return(x)
}

obesity = binning(obesity)

```

As we saw with the “str()” function, all the categorical variables are treated as “character”. Therefore, we will convert all the categorical variables to “factor” type.

Just as we did with the binning, we created a function to convert character variables to factor. This function is called “to_factor”.

```
# Converting character variables to factor :

to_factor <- function(x) {

  x$Gender = as.factor(x$Gender)
  x$family_history = as.factor(x$family_history)
  x$eat_caloric = as.factor(x$eat_caloric)
  x$food_inbetween = as.factor(x$food_inbetween)
  x$SMOKE = as.factor(x$SMOKE)
  x$monitor_cal = as.factor(x$monitor_cal)
  x$alcohol = as.factor(x$alcohol)
  x$MTRANS = as.factor(x$MTRANS)
  x$NObeyesdad = as.factor(x$NObeyesdad)
  x$vegetables = as.factor(x$vegetables)
  x$main_meals = as.factor(x$main_meals)
  x$CH20 = as.factor(x$CH20)
  x$physical_act = as.factor(x$physical_act)
  x$tech_devices = as.factor(x$tech_devices)

  return(x)

}

obesity = to_factor(obesity)
```

Our next step will to remove any missing values.

```
# Checking if there are Missing Values :

sum(is.na(obesity))
```

```
## [1] 0
```

There are no missing values within our dataset.

We will now proceed with the dummification of the categorical variables. All variables (with the exception of gender, age, height and weight) have already been dummified.


```

# Dummyfing the binary
# variables(family_history, eat_caloric,
# SMOKE, and monitor_cal) :

# Gender 1 = female, 0 = male
obesity_dummy <- cbind(dummy(obesity$Gender, sep = "_"),
  obesity[2:17])

```

```

## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored

```

```

names(obesity_dummy)[1] <- c("Gender")
obesity_dummy <- subset(obesity_dummy, select = -c(2))

# family_history 1 = yes, 0 = no
obesity_dummy <- cbind(obesity_dummy[1:4], dummy(obesity_dummy$family_hist,
  sep = "_"), obesity_dummy[6:17])

```

```

## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored

```

```

names(obesity_dummy)[6] <- c("family_hist")
obesity_dummy <- subset(obesity_dummy, select = -c(5))

# eat_caloric with 1 = yes, 0 = no
obesity_dummy <- cbind(obesity_dummy[1:5], dummy(obesity_dummy$eat_caloric,
  sep = "_"), obesity_dummy[7:17])

```

```

## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored

```

```

names(obesity_dummy)[7] <- c("eat_caloric")
obesity_dummy <- subset(obesity_dummy, select = -c(6))

# SMOKE 1 = yes, 0 = no
obesity_dummy <- cbind(obesity_dummy[1:9], dummy(obesity_dummy$SMOKE,
  sep = "_"), obesity_dummy[11:17])

```

```

## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored

```

```

names(obesity_dummy)[11] <- c("smoke")
obesity_dummy <- subset(obesity_dummy, select = -c(10))

# monitor_cal 1 = yes, 0 = no
obesity_dummy <- cbind(obesity_dummy[1:11], dummy(obesity_dummy$monitor_cal,
  sep = "_"), obesity_dummy[13:17])

## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored

names(obesity_dummy)[13] <- c("monitor_cal")
obesity_dummy <- subset(obesity_dummy, select = -c(12))

# Dummyfying the categorical variables

# vegetables
obesity_dum <- cbind(obesity_dummy[1:6], dummy(obesity_dummy$vegetables,
  sep = "_"), obesity_dummy[8:17])

## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored

names(obesity_dum)[7:9] <- c("vegetables_never",
  "vegetables_sometimes", "vegetables_always")

# main_meals
obesity_dum <- cbind(obesity_dum[1:9], dummy(obesity_dum$main_meals,
  sep = "_"), obesity_dum[11:19])

## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored

names(obesity_dum)[10:12] <- c("main_meals_Btw_1_2",
  "main_meals_More_than_3", "main_meals_three")

# food_in_between
obesity_dum <- cbind(obesity_dum[1:12], dummy(obesity_dum$food_inbetween,
  sep = "_"), obesity_dum[14:21])

## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored

```

```
names(obesity_dum)[13:16] <- c("food_inbetween_always",
  "food_inbetween_frequently", "food_inbetween_no",
  "food_inbetween_sometimes")
```

```
# alcohol
```

```
obesity_dum <- cbind(obesity_dum[1:21], dummy(obesity_dum$alcohol,
  sep = "_"), obesity_dum[23:24])
```

```
## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored
```

```
names(obesity_dum)[22:25] <- c("alcohol_always",
  "alcohol_frequently", "alcohol_no", "alcohol_sometimes")
```

```
# MTRANS
```

```
obesity_dum <- cbind(obesity_dum[1:25], dummy(obesity_dum$MTRANS,
  sep = "_"), obesity_dum[27])
```

```
## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored
```

```
names(obesity_dum)[26:30] <- c("mtrans_automobile",
  "mtrans_bike", "mtrans_motorbike", "mtrans_public_transportation",
  "mtrans_walking")
```

```
# CH2O
```

```
obesity_dum <- cbind(obesity_dum[1:17], dummy(obesity_dum$CH2O,
  sep = "_"), obesity_dum[19:31])
```

```
## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored
```

```
names(obesity_dum)[18:20] <- c("CH2O_less_than_a_liter",
  "CH2O_between_1_and_2", "CH2O_more_than_2")
```

```
# physical_act
```

```
obesity_dum <- cbind(obesity_dum[1:21], dummy(obesity_dum$physical_act,
  sep = "_"), obesity_dum[23:33])
```

```
## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored
```

```
names(obesity_dum)[22:24] <- c("physical_act_do_not_have",
  "physical_act_1_2", "physical_act_2_4")

# tech_devices : this one is a little bit
# tricky since there are many categories but
# only one is represented within the data!

obesity_dum <- cbind(obesity_dum[1:24], dummy(obesity_dum$tech_devices,
  sep = "_"), obesity_dum[26:35])
```

```
## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored
```

```
names(obesity_dum)[25:27] <- c("tech_0_hours",
  "tech_1_hour", "tech_2_hours_or_more")

# remove(obesity_dum)
obesity_dum <- subset(obesity_dum[c(1:36)])
```

Finally, the last step in the data pre-processing is the partitioning of the data. We partitioned the data into a 60% training set and a 40% validation set. Because we have a relatively small number of observations (only 2111 observations), we thought it best to exclude a test set. However, better results could be obtained if we kept a third “test set”.

```
# Partitioning the data (60% training, 40%
# validation)

set.seed(1)

train.obs <- sample(rownames(obesity_dum), dim(obesity_dum)[1] *
  0.6)
train.set <- obesity_dum[train.obs, ]

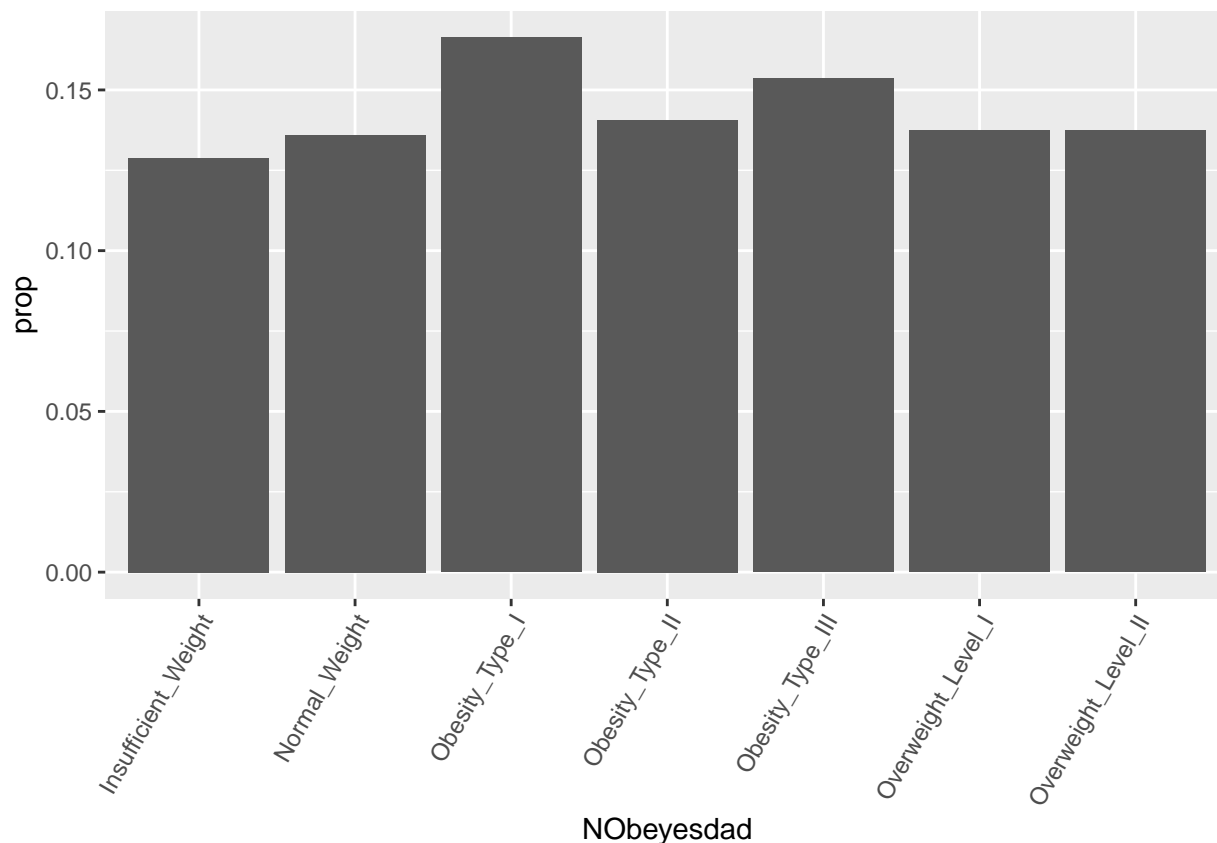
set.seed(1)

valid.obs <- setdiff(rownames(obesity_dum), train.obs)
valid.set <- obesity_dum[valid.obs, ]
```

Now that we have finished with the data pre-processing, we can proceed with the exploratory data analysis. While we have dummified variables in the steps above, the original non-dummified versions of the variables will be used in the exploratory data analysis for visualization purposes.

Exploratory Data Analysis

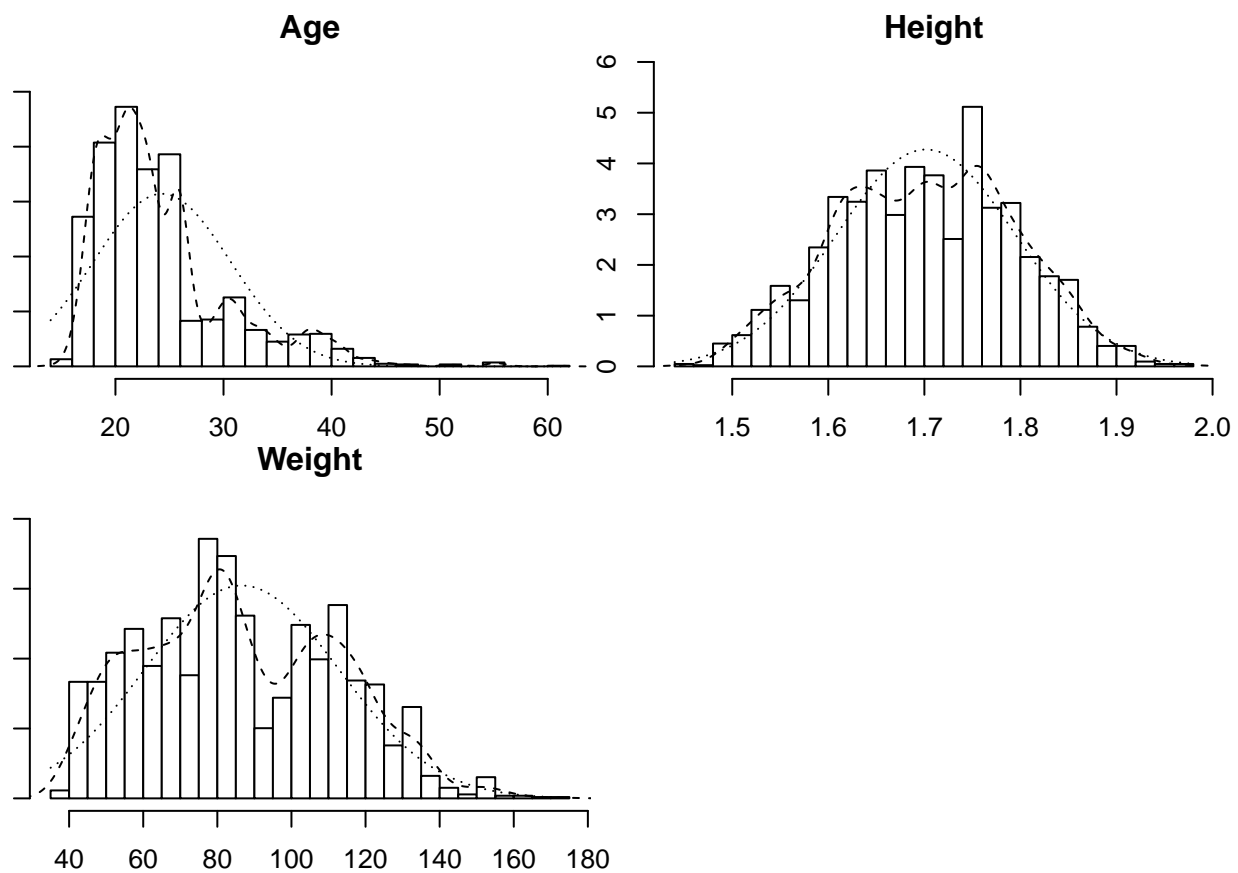
```
ggplot(data = obesity, aes(x = NObeyesdad)) +  
  geom_bar(aes(y = ..prop.., group = 1)) + theme(axis.text.x = element_text(angle = 60,  
  hjust = 1))
```



We see that the distribution of observations across the different weights is quite uniform, meaning that we do not have an unbalanced data set with respect to our variable of interest (the weight).

Let's now look at some histograms for all the continuous variables in our dataset.

```
# Creating histograms :  
multi.hist(obesity[, 2:4], density = TRUE)
```



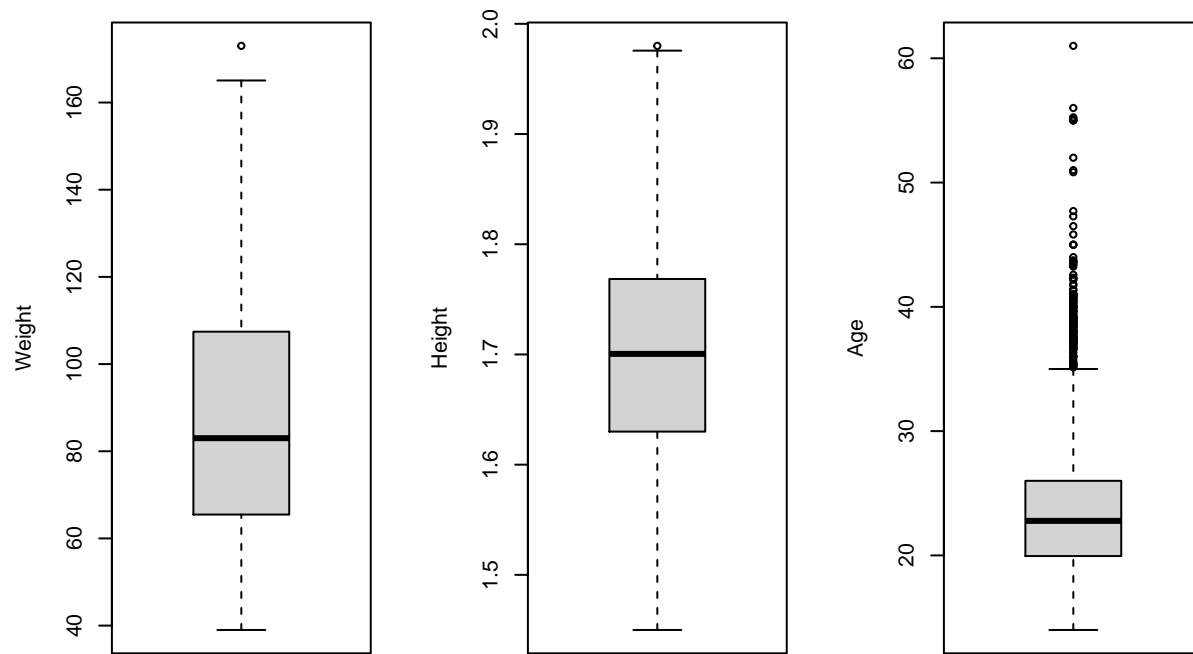
Creating boxplots :

```
par(mfrow = c(1, 3))
```

```
boxplot(obesity$Weight, ylab = "Weight")
```

```
boxplot(obesity$Height, ylab = "Height")
```

```
boxplot(obesity$Age, ylab = "Age")
```



```
# We may have ONE outlier for Weight, and
# almost one for Height! Age is VERY right
# skewed!
```

```
# comment on why we are not removing the
# outliers
```

Interpretation:

...

Now, let's do some barplots in order to get an idea of the distribution of each of the categorical variables.

```
# Barplots :
```

```
plot_1 = ggplot(data = obesity, aes(x = NObeyesdad)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)
```

```
plot_2 = ggplot(data = obesity, aes(x = main_meals)) +
  geom_bar(aes(y = ..count.., group = 1)) +
```

```

    theme(axis.text.x = element_text(angle = 60,
      hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_3 = ggplot(data = obesity, aes(x = Gender)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_4 = ggplot(data = obesity, aes(x = family_history)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_5 = ggplot(data = obesity, aes(x = vegetables)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_6 = ggplot(data = obesity, aes(x = food_inbetween)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_7 = ggplot(data = obesity, aes(x = tech_devices)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_8 = ggplot(data = obesity, aes(x = eat_caloric)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_9 = ggplot(data = obesity, aes(x = SMOKE)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_10 = ggplot(data = obesity, aes(x = CH20)) +

```



```

geom_bar(aes(y = ..count.., group = 1)) +
theme(axis.text.x = element_text(angle = 60,
  hjust = 1)) + geom_text(stat = "count",
aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_11 = ggplot(data = obesity, aes(x = monitor_cal)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_12 = ggplot(data = obesity, aes(x = physical_act)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

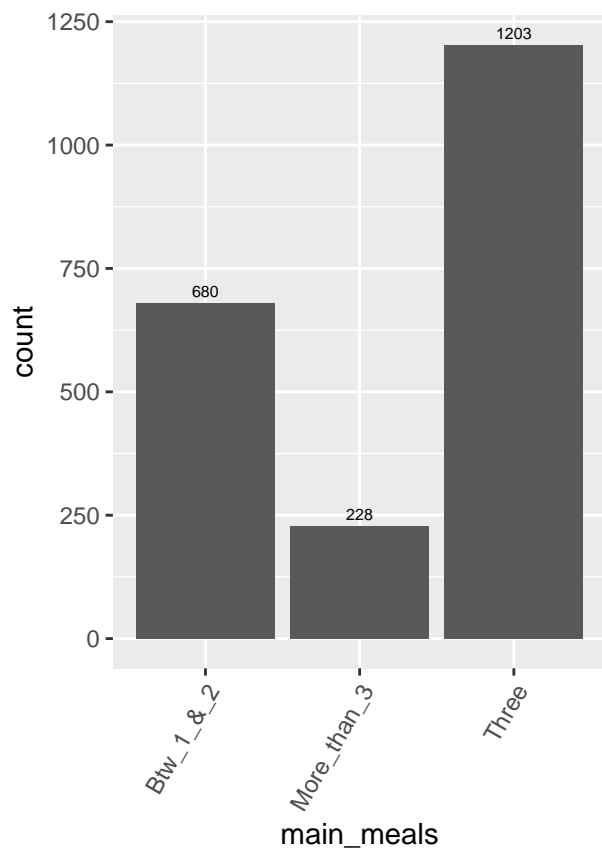
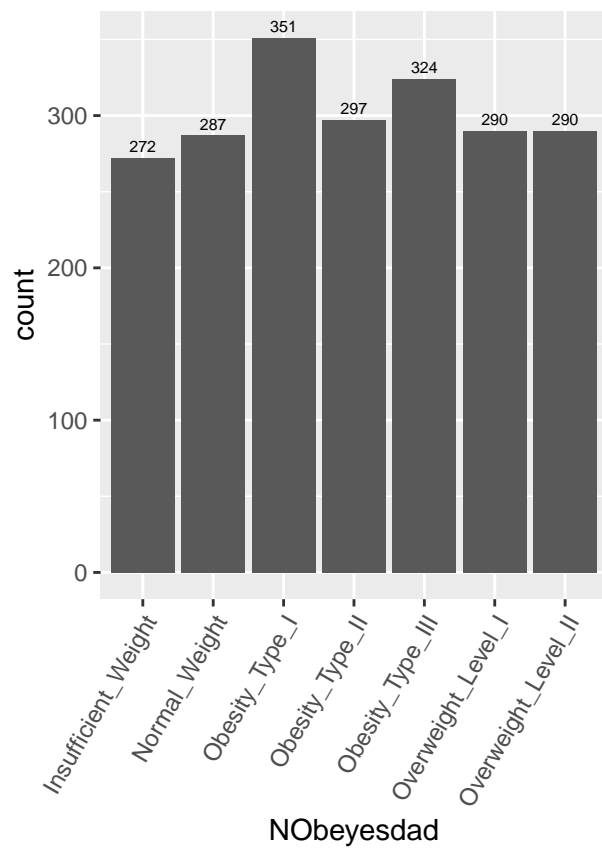
plot_13 = ggplot(data = obesity, aes(x = alcohol)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

plot_14 = ggplot(data = obesity, aes(x = MTRANS)) +
  geom_bar(aes(y = ..count.., group = 1)) +
  theme(axis.text.x = element_text(angle = 60,
    hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

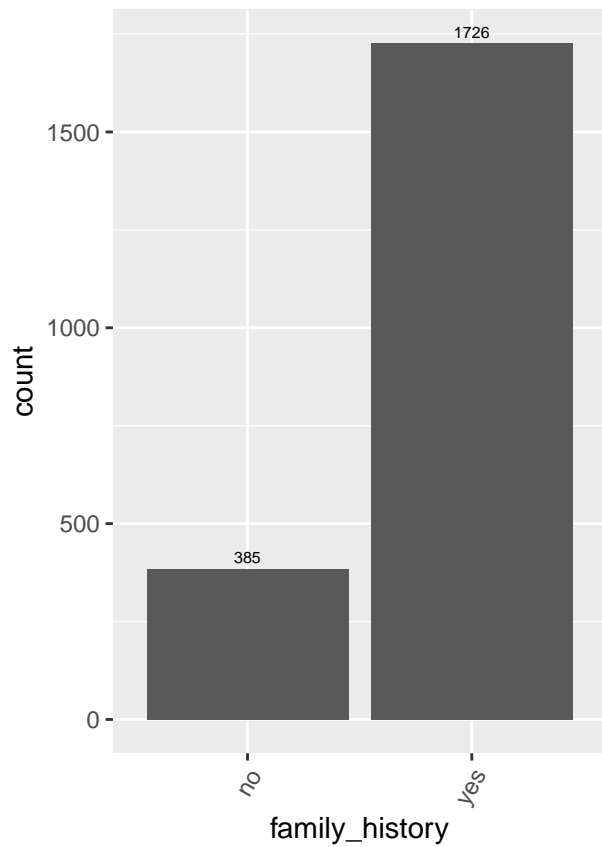
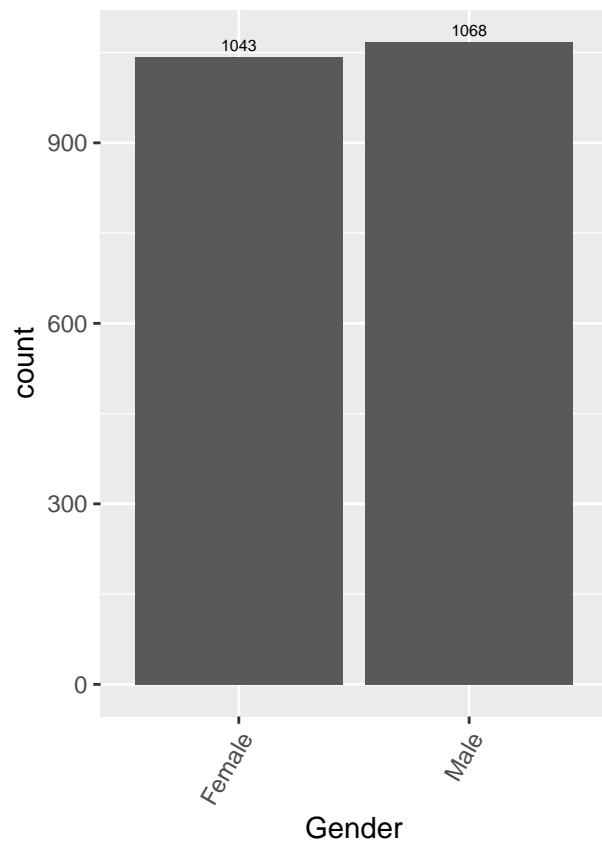
# Arranging them two-by-two :

grid.arrange(plot_1, plot_2, ncol = 2)

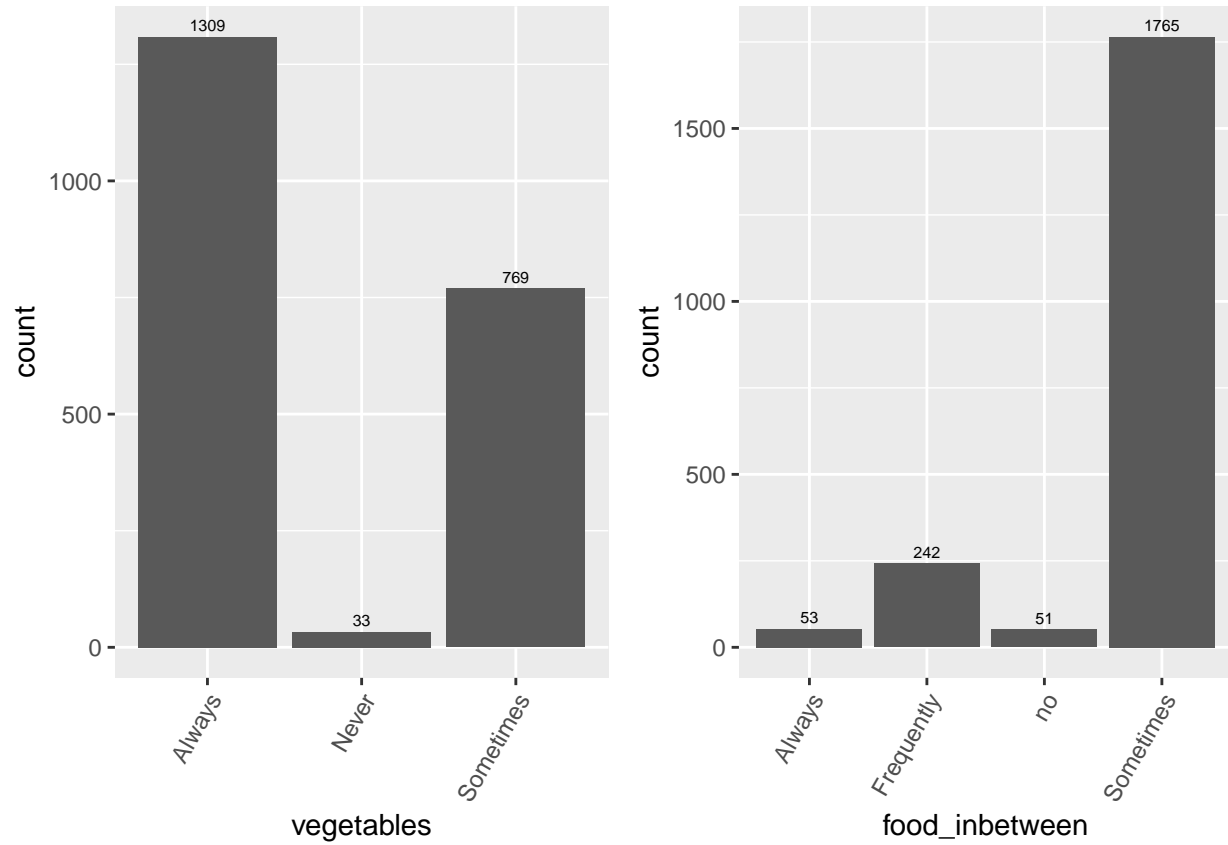
```



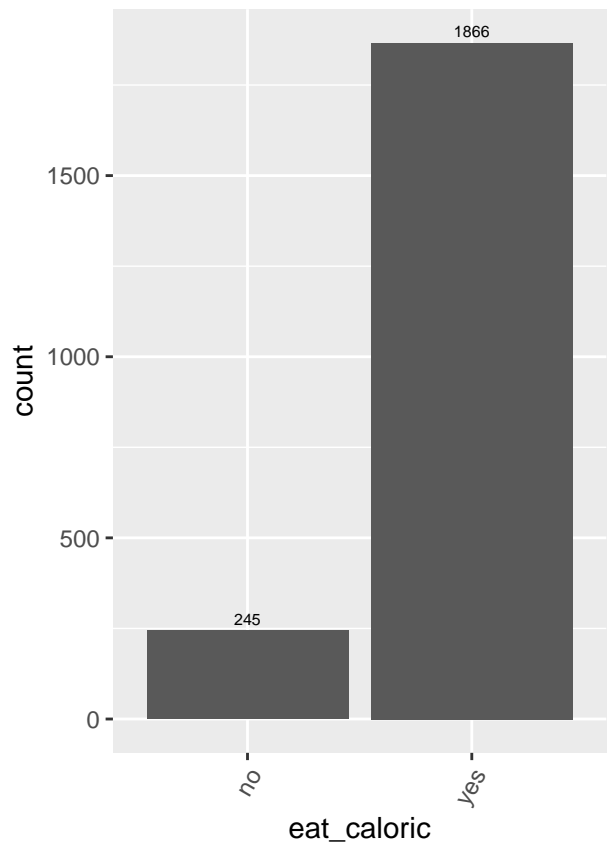
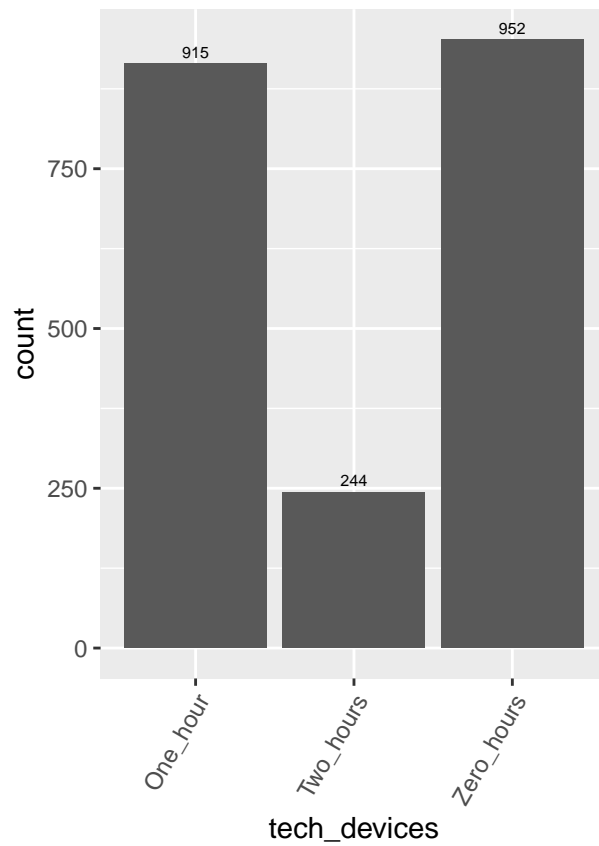
```
grid.arrange(plot_3, plot_4, ncol = 2)
```



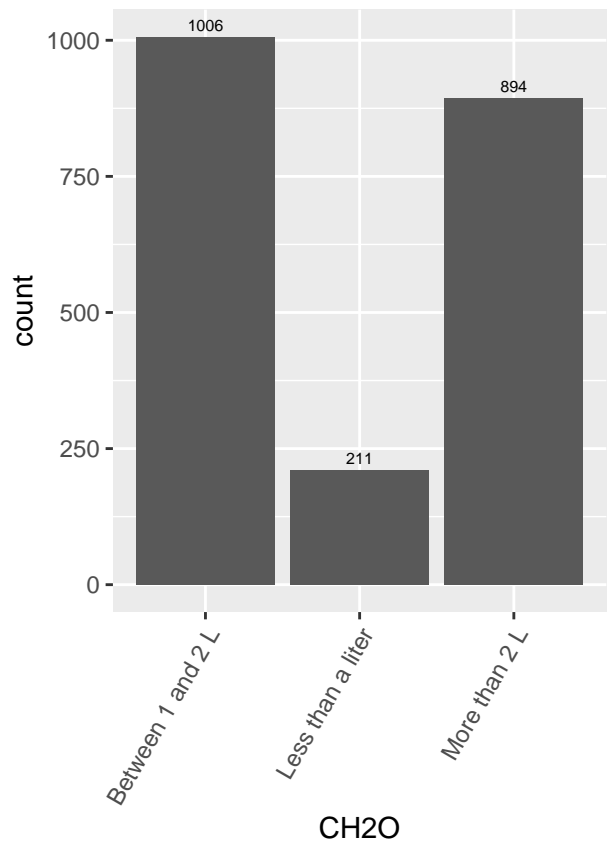
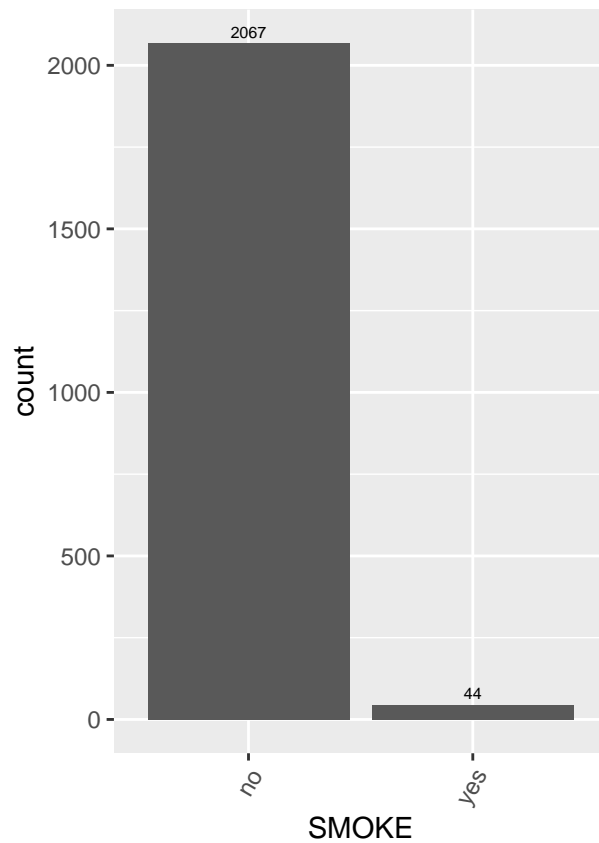
```
grid.arrange(plot_5, plot_6, ncol = 2)
```



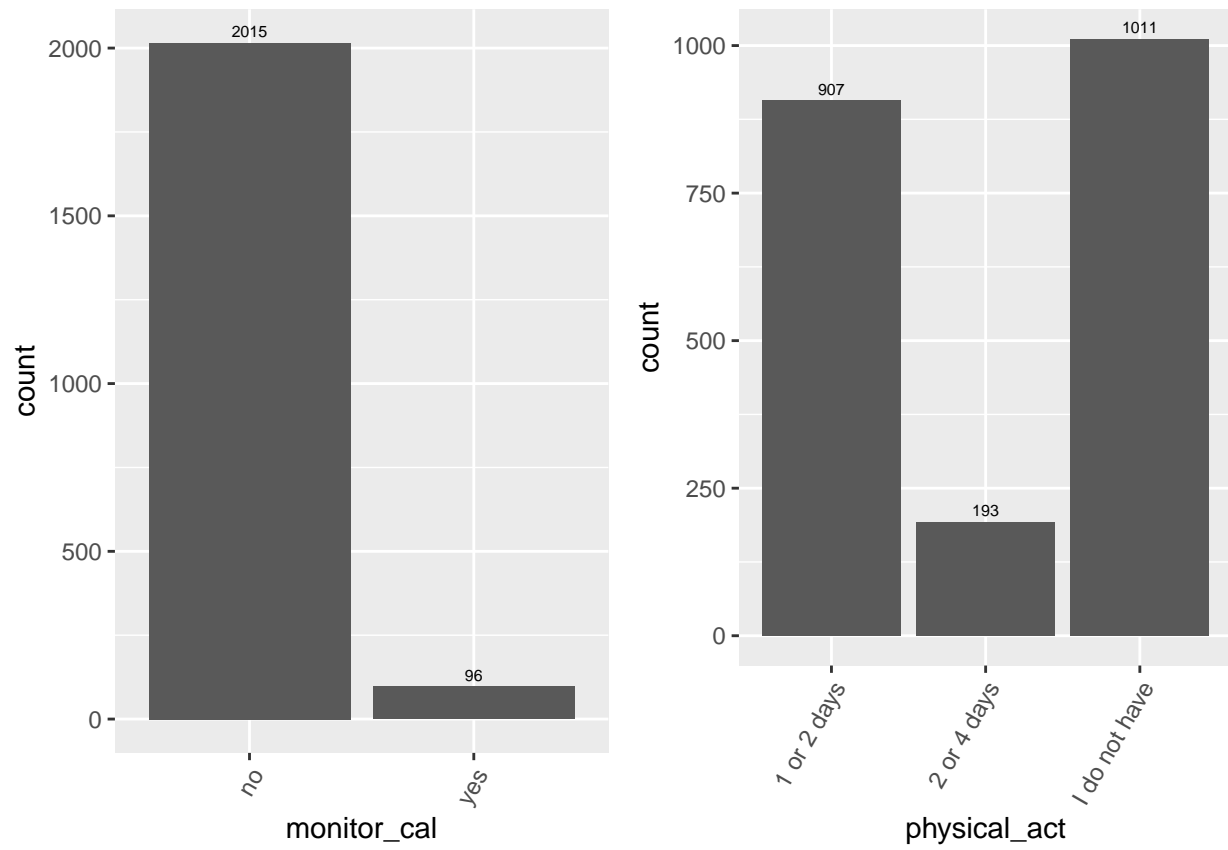
```
grid.arrange(plot_7, plot_8, ncol = 2)
```



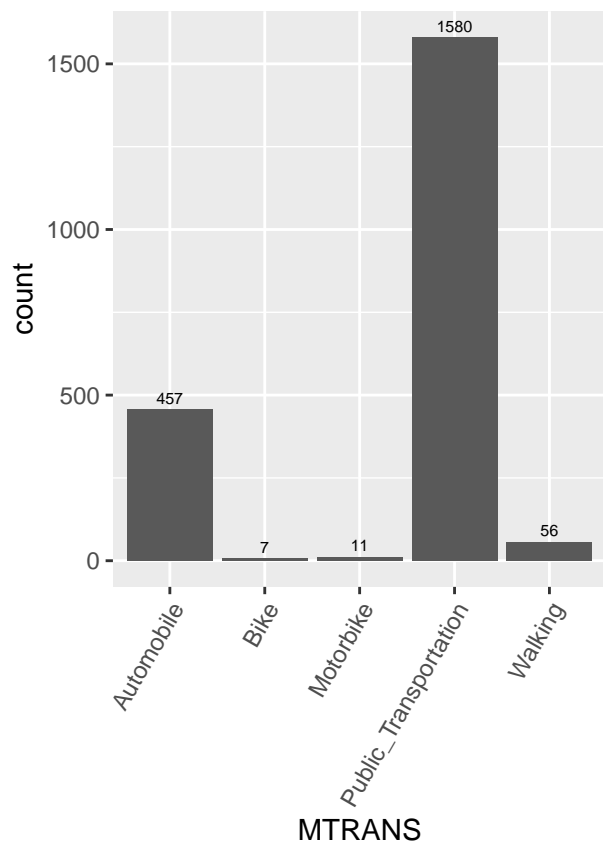
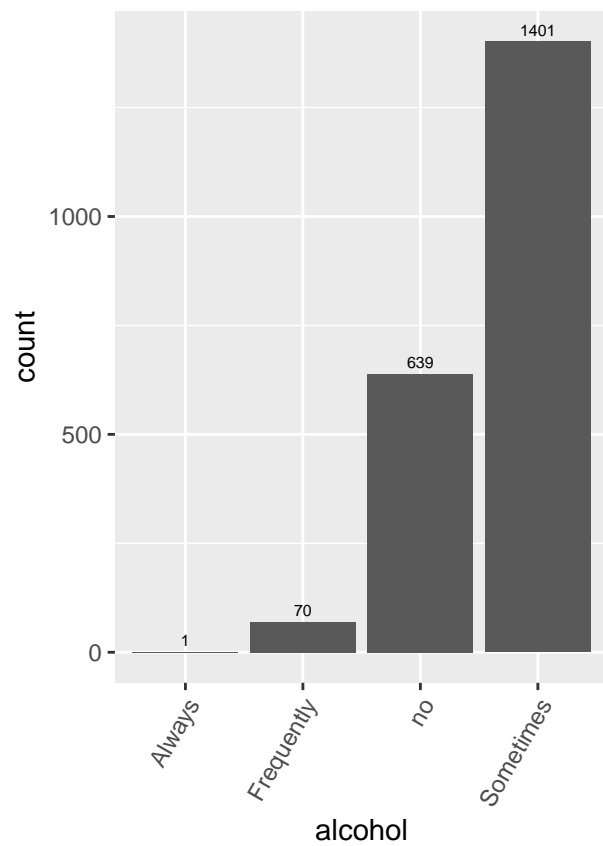
```
grid.arrange(plot_9, plot_10, ncol = 2)
```



```
grid.arrange(plot_11, plot_12, ncol = 2)
```

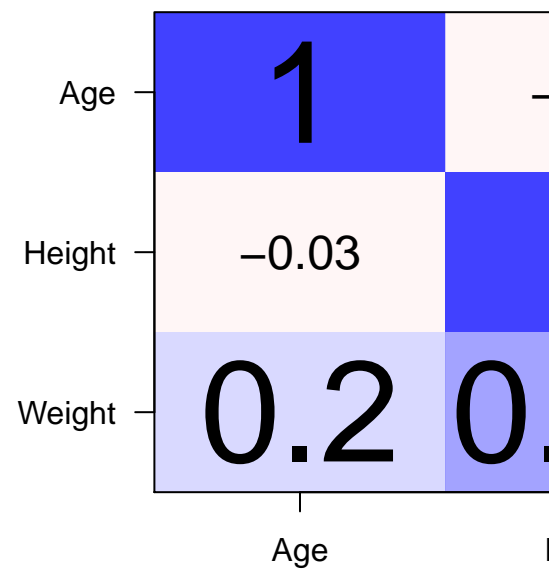


```
grid.arrange(plot_13, plot_14, ncol = 2)
```



Correlation plot

```
cor.plot(na.omit(obesity[c(2, 3, 4)]))
```



Let's look at the correlations between the variables.

```
# comment on correlation and how we didnt do  
# it for categorical variables
```

Data Analysis

Multiple Linear Regression

We begin with a multiple linear regression model. We will first run a full model with (n-1) dummy categories included for each variable. In most cases the dummy that was excluded from the formula this the dummy which referred to the variable category “no” or equivalent. For instance, for the variable alcohol, we excluded the variable alcohol_no from the model formula.

```
# Linear regression  
  
# formula: Weight = Gender, Age, Height,  
  
lm_weight <- lm(Weight ~ Gender + Age + Height +  
  family_hist + eat_caloric + vegetables_sometimes +  
  vegetables_always + main_meals_Btw_1_2 + main_meals_More_than_3 +
```

```

food_inbetween_always + food_inbetween_frequently +
food_inbetween_sometimes + smoke + CH20_between_1_and_2 +
CH20_more_than_2 + monitor_cal + physical_act_1_2 +
physical_act_2_4 + tech_1_hour + tech_2_hours_or_more +
alcohol_always + alcohol_frequently + alcohol_sometimes +
mtrans_automobile + mtrans_bike + mtrans_public_transportation,
data = train.set)

```

```
summary(lm_weight)
```

```

##
## Call:
## lm(formula = Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##     vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##     main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##     food_inbetween_sometimes + smoke + CH20_between_1_and_2 +
##     CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
##     tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_frequently +
##     alcohol_sometimes + mtrans_automobile + mtrans_bike + mtrans_public_transportatio
##     data = train.set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -54.921  -9.621   0.615   9.564  54.196
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -156.56957    13.02841  -12.018  < 2e-16 ***
## Gender              4.25524     1.24684   3.413  0.000664 ***
## Age                0.81274     0.09894   8.214  5.32e-16 ***
## Height           121.80520     7.14305  17.052  < 2e-16 ***
## family_hist       15.29655     1.32971  11.504  < 2e-16 ***
## eat_caloric        3.96819     1.47926   2.683  0.007404 **
## vegetables_sometimes -9.52226     3.44567  -2.764  0.005802 **
## vegetables_always  -7.32851     0.99765  -7.346  3.70e-13 ***
## main_meals_Btw_1_2  -5.50831     1.04130  -5.290  1.45e-07 ***
## main_meals_More_than_3 -18.03950     1.53632 -11.742  < 2e-16 ***
## food_inbetween_always  -3.16154     4.25233  -0.743  0.457330
## food_inbetween_frequently -17.21409     3.42556  -5.025  5.77e-07 ***
## food_inbetween_sometimes  0.57720     3.22714   0.179  0.858080
## smoke             -0.22053     3.16589  -0.070  0.944476
## CH20_between_1_and_2  -0.52713     1.59331  -0.331  0.740821
## CH20_more_than_2      5.35807     0.98999   5.412  7.47e-08 ***
## monitor_cal        -4.92421     2.24731  -2.191  0.028626 *
## physical_act_1_2     -9.38751     1.63586  -5.739  1.20e-08 ***
## physical_act_2_4      2.02866     1.00094   2.027  0.042901 *
## tech_1_hour         -5.45663     1.55983  -3.498  0.000485 ***

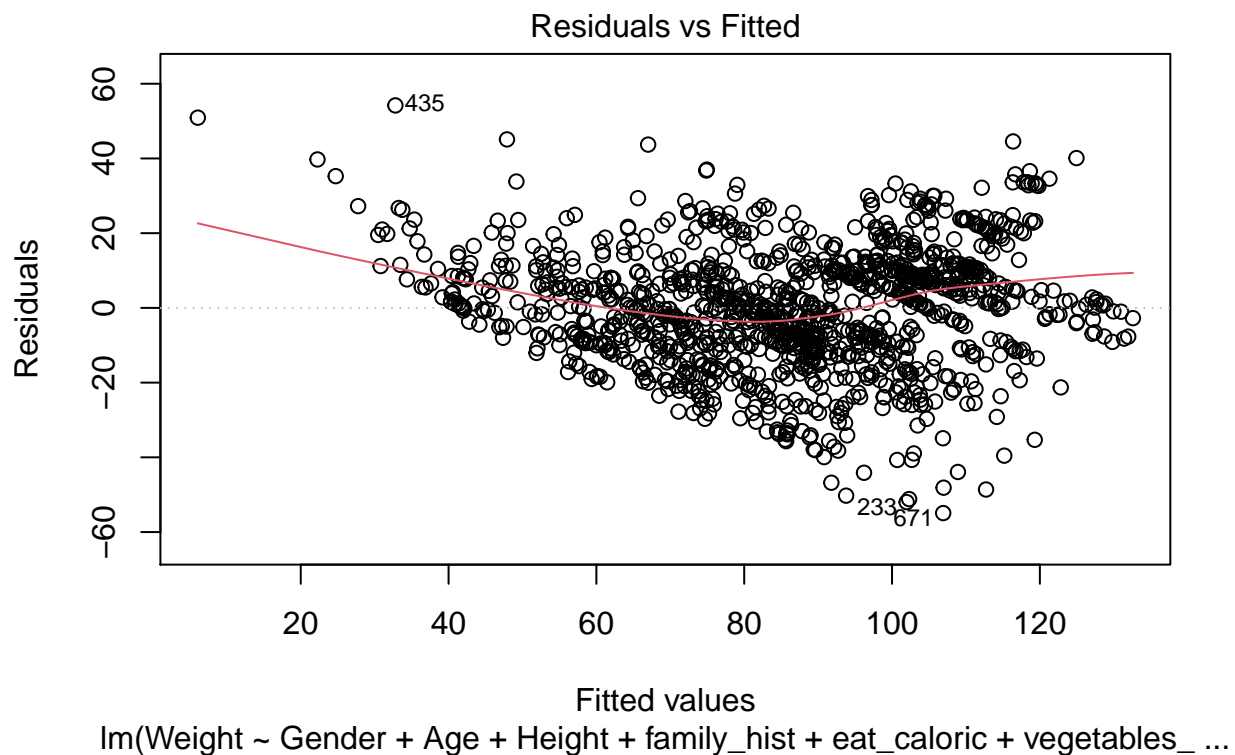
```

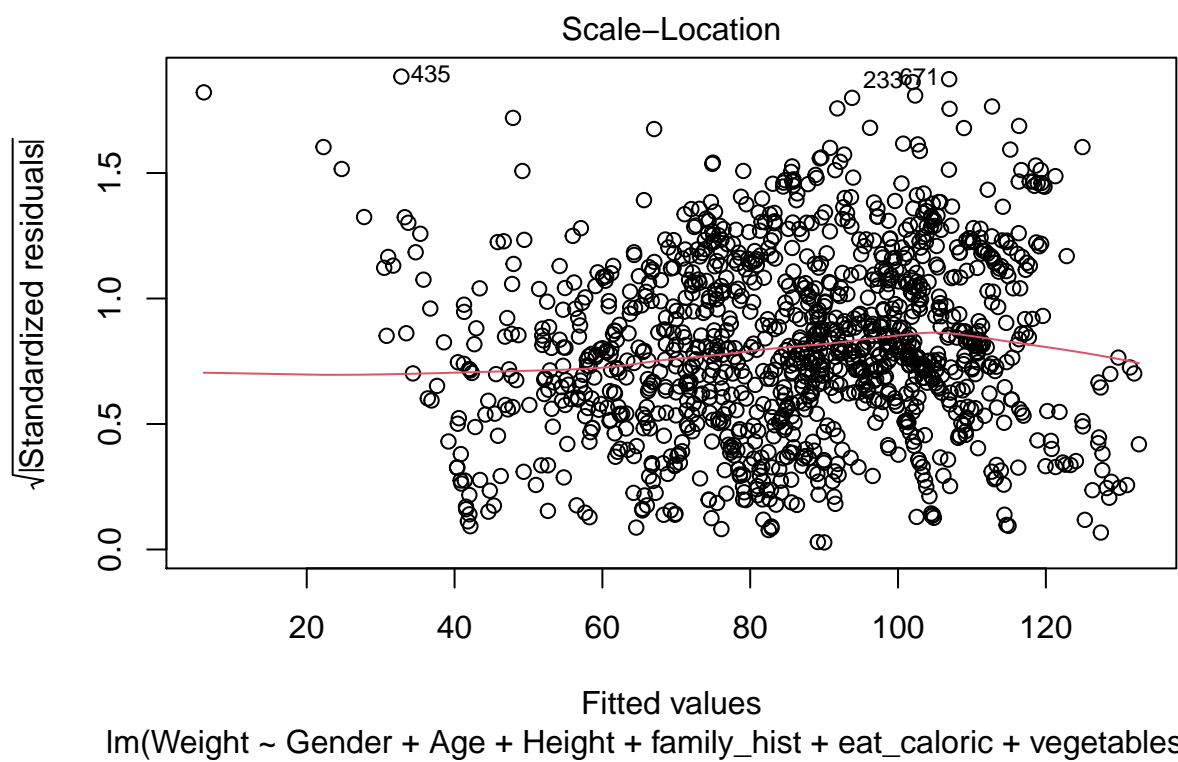
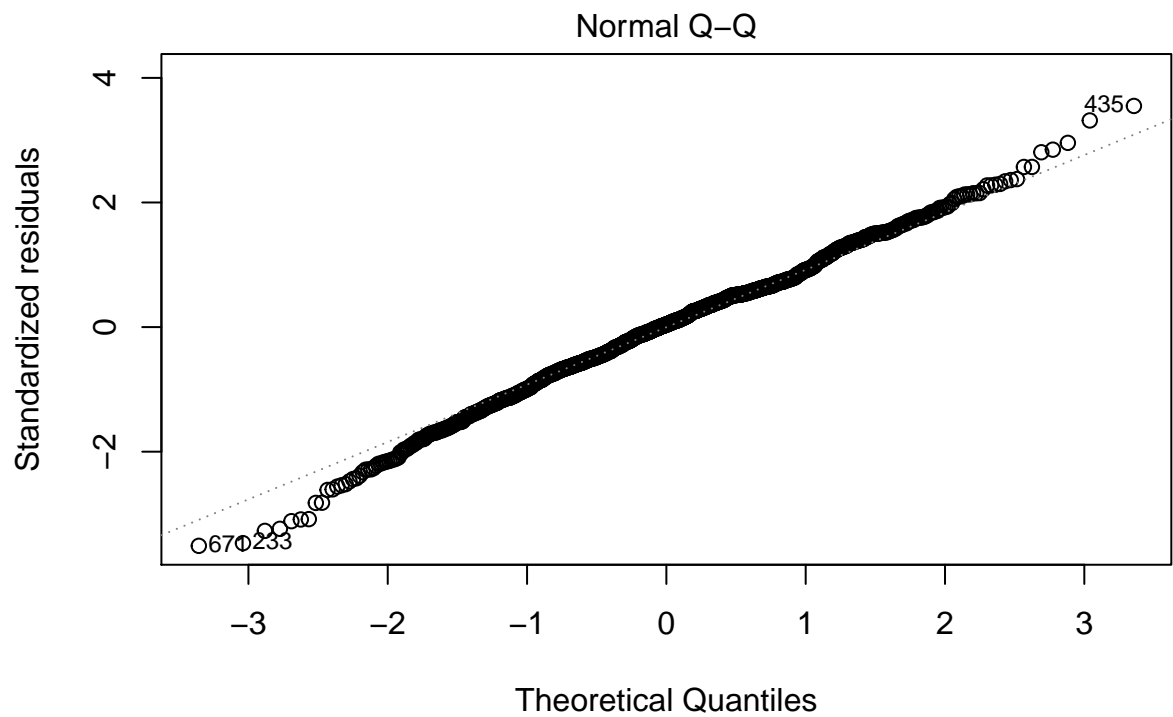


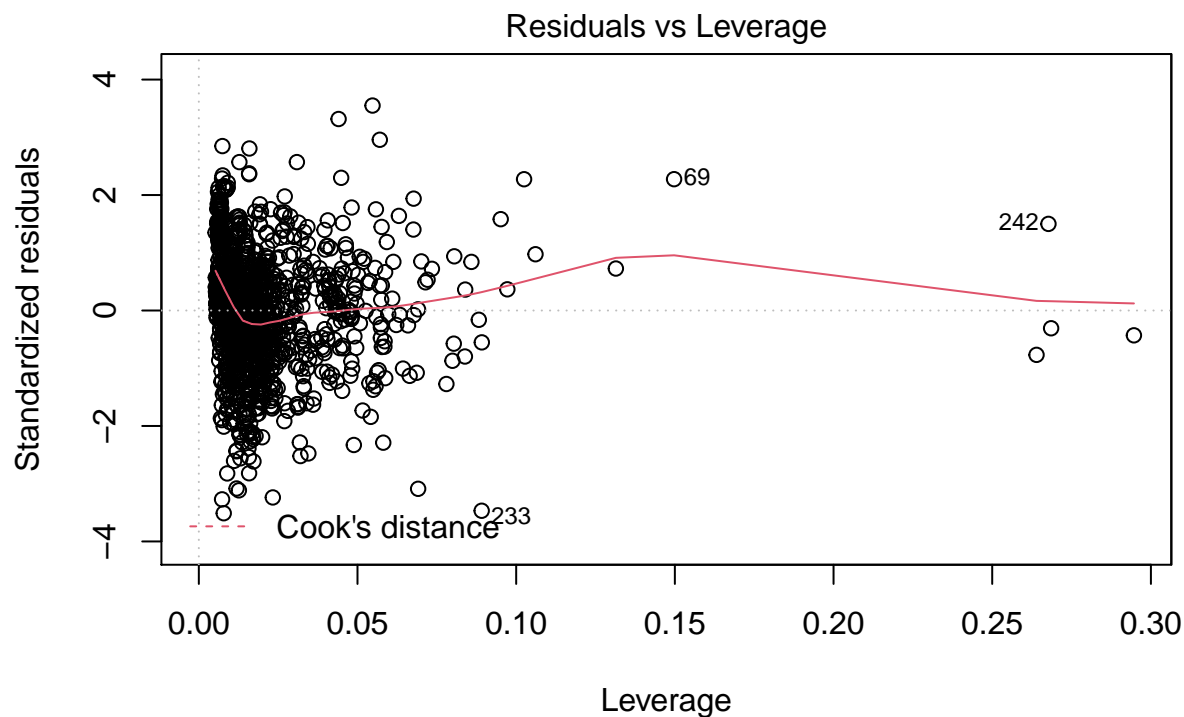
```
## tech_2_hours_or_more      -1.47184      1.00579    -1.463  0.143621
## alcohol_always            13.67598     16.05482      0.852  0.394473
## alcohol_frequently        -1.19652      2.63648     -0.454  0.650030
## alcohol_sometimes          4.61559      1.04280      4.426  1.04e-05 ***
## mtrans_automobile         -7.08215      2.83046     -2.502  0.012473 *
## mtrans_bike                -4.04084      8.28314     -0.488  0.625750
## mtrans_public_transportation 4.55777      2.59702      1.755  0.079506 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.7 on 1239 degrees of freedom
## Multiple R-squared:  0.6464, Adjusted R-squared:  0.639
## F-statistic: 87.11 on 26 and 1239 DF,  p-value: < 2.2e-16
```

```
plot(lm_weight)
```

```
## Warning: not plotting observations with leverage one:
##      768
```







lm(Weight ~ Gender + Age + Height + family_hist + eat_caloric + vegetables_ ...

Looking at the model above, we have quite a lot of variables that are significant at a confidence level of 95%. The variables that are not significant are: food_inbetween_always, food_inbetween_sometimes, smoke, CH2O_between_1_and_2, tech_1_hour, alcohol_always, alcohol_frequently, mtrans_bike and mtrans_public_transportation.

Because there are many significant variables, we will not interpret all of them, instead, we will interpret some that we find interesting.

- **Age:** An increase of 1 year of age corresponds to an average increase of 0.812 kg in weight, ceteris paribus.
- **main_meals_Btw_1_2:** An individual that eats between 1 and 2 main meals per day has an average decrease of 5.508 kg in comparison to an individual that eats three main meals per day.

Because we wish to select the best possible model for the linear regression, we will proceed with the stepwise selection method, in order to choose the most appropriate one. We will run a forward, backward, and both model selection.

```
# Stepwise model selection
```

```
# Forward
```

```
lm_forward_obesity <- step(lm_weight, direction = "forward")
```

```
## Start: AIC=6999.41
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##     vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##     main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##     food_inbetween_sometimes + smoke + CH20_between_1_and_2 +
##     CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
##     tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_frequently +
##     alcohol_sometimes + mtrans_automobile + mtrans_bike + mtrans_public_transportatio
```

```
summary(lm_forward_obesity)
```

```
##
## Call:
## lm(formula = Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##     vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##     main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##     food_inbetween_sometimes + smoke + CH20_between_1_and_2 +
##     CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
##     tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_frequently +
##     alcohol_sometimes + mtrans_automobile + mtrans_bike + mtrans_public_transportatio
##     data = train.set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -54.921  -9.621   0.615   9.564  54.196
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -156.56957    13.02841  -12.018  < 2e-16 ***
## Gender              4.25524     1.24684   3.413 0.000664 ***
## Age                0.81274     0.09894   8.214 5.32e-16 ***
## Height           121.80520     7.14305  17.052  < 2e-16 ***
## family_hist       15.29655     1.32971  11.504  < 2e-16 ***
## eat_caloric        3.96819     1.47926   2.683 0.007404 **
## vegetables_sometimes -9.52226     3.44567  -2.764 0.005802 **
## vegetables_always  -7.32851     0.99765  -7.346 3.70e-13 ***
## main_meals_Btw_1_2  -5.50831     1.04130  -5.290 1.45e-07 ***
## main_meals_More_than_3 -18.03950     1.53632 -11.742  < 2e-16 ***
## food_inbetween_always  -3.16154     4.25233  -0.743 0.457330
## food_inbetween_frequently -17.21409     3.42556  -5.025 5.77e-07 ***
## food_inbetween_sometimes  0.57720     3.22714   0.179 0.858080
## smoke            -0.22053     3.16589  -0.070 0.944476
## CH20_between_1_and_2  -0.52713     1.59331  -0.331 0.740821
## CH20_more_than_2      5.35807     0.98999   5.412 7.47e-08 ***
## monitor_cal        -4.92421     2.24731  -2.191 0.028626 *
## physical_act_1_2     -9.38751     1.63586  -5.739 1.20e-08 ***
## physical_act_2_4      2.02866     1.00094   2.027 0.042901 *
```

```
## tech_1_hour -5.45663 1.55983 -3.498 0.000485 ***
## tech_2_hours_or_more -1.47184 1.00579 -1.463 0.143621
## alcohol_always 13.67598 16.05482 0.852 0.394473
## alcohol_frequently -1.19652 2.63648 -0.454 0.650030
## alcohol_sometimes 4.61559 1.04280 4.426 1.04e-05 ***
## mtrans_automobile -7.08215 2.83046 -2.502 0.012473 *
## mtrans_bike -4.04084 8.28314 -0.488 0.625750
## mtrans_public_transportation 4.55777 2.59702 1.755 0.079506 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.7 on 1239 degrees of freedom
## Multiple R-squared: 0.6464, Adjusted R-squared: 0.639
## F-statistic: 87.11 on 26 and 1239 DF, p-value: < 2.2e-16
```

```
# AIC: 6999.41 Model : Weight ~ Gender + Age +
# Height + family_hist + eat_caloric +
# vegetables_sometimes + vegetables_always +
# main_meals_Btw_1_2 + main_meals_More_than_3
# + food_inbetween_always +
# food_inbetween_frequently +
# food_inbetween_sometimes + smoke +
# CH20_between_1_and_2 + CH20_more_than_2 +
# monitor_cal + physical_act_1_2 +
# physical_act_2_4 + tech_1_hour +
# tech_2_hours_or_more + alcohol_always +
# alcohol_frequently + alcohol_sometimes +
# mtrans_automobile + mtrans_bike +
# mtrans_public_transportation
```

```
# Backward
```

```
lm_backward_obesity <- step(lm_weight, direction = "backward")
```

```
## Start: AIC=6999.41
```

```
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
## vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
## main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
## food_inbetween_sometimes + smoke + CH20_between_1_and_2 +
## CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
## tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_frequently +
## alcohol_sometimes + mtrans_automobile + mtrans_bike + mtrans_public_transportation
```

```
##
## Df Sum of Sq RSS AIC
## - smoke 1 1 305507 6997.4
## - food_inbetween_sometimes 1 8 305514 6997.4
## - CH20_between_1_and_2 1 27 305533 6997.5
```

```

## - alcohol_frequently      1      51 305556 6997.6
## - mtrans_bike             1      59 305564 6997.7
## - food_inbetween_always   1     136 305642 6998.0
## - alcohol_always          1     179 305685 6998.2
## <none>                    305506 6999.4
## - tech_2_hours_or_more    1     528 306034 6999.6
## - mtrans_public_transportation 1     759 306265 7000.6
## - physical_act_2_4        1    1013 306518 7001.6
## - monitor_cal             1    1184 306689 7002.3
## - mtrans_automobile        1    1544 307049 7003.8
## - eat_caloric              1    1774 307280 7004.7
## - vegetables_sometimes     1    1883 307389 7005.2
## - Gender                   1    2872 308378 7009.3
## - tech_1_hour              1    3017 308523 7009.9
## - alcohol_sometimes        1    4831 310336 7017.3
## - food_inbetween_frequently 1    6227 311732 7023.0
## - main_meals_Btw_1_2       1    6900 312405 7025.7
## - CH20_more_than_2         1    7223 312728 7027.0
## - physical_act_1_2         1    8120 313626 7030.6
## - vegetables_always        1   13305 318811 7051.4
## - Age                      1   16637 322143 7064.5
## - family_hist              1   32630 338136 7125.9
## - main_meals_More_than_3    1   33997 339502 7131.0
## - Height                   1   71699 377204 7264.3
##
## Step:  AIC=6997.41
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##      vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##      main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##      food_inbetween_sometimes + CH20_between_1_and_2 + CH20_more_than_2 +
##      monitor_cal + physical_act_1_2 + physical_act_2_4 + tech_1_hour +
##      tech_2_hours_or_more + alcohol_always + alcohol_frequently +
##      alcohol_sometimes + mtrans_automobile + mtrans_bike + mtrans_public_transportation
##
##              Df Sum of Sq    RSS    AIC
## - food_inbetween_sometimes      1      8 305515 6995.4
## - CH20_between_1_and_2           1     27 305534 6995.5
## - alcohol_frequently             1     53 305560 6995.6
## - mtrans_bike                   1     58 305565 6995.7
## - food_inbetween_always          1    136 305643 6996.0
## - alcohol_always                 1    179 305686 6996.2
## <none>                          305507 6997.4
## - tech_2_hours_or_more           1    527 306034 6997.6
## - mtrans_public_transportation    1    760 306267 6998.6
## - physical_act_2_4               1   1012 306519 6999.6
## - monitor_cal                   1   1194 306701 7000.4
## - mtrans_automobile              1   1547 307054 7001.8
## - eat_caloric                   1   1776 307283 7002.8

```

```

## - vegetables_sometimes      1      1884 307391 7003.2
## - Gender                    1      2871 308378 7007.3
## - tech_1_hour               1      3030 308537 7007.9
## - alcohol_sometimes         1      4830 310336 7015.3
## - food_inbetween_frequently 1      6234 311741 7021.0
## - main_meals_Btw_1_2        1      6901 312408 7023.7
## - CH20_more_than_2          1      7315 312822 7025.4
## - physical_act_1_2          1      8119 313626 7028.6
## - vegetables_always         1     13310 318817 7049.4
## - Age                       1     17080 322587 7064.3
## - family_hist               1     32631 338138 7123.9
## - main_meals_More_than_3     1     34024 339531 7129.1
## - Height                    1     72330 377837 7264.4
##
## Step:  AIC=6995.45
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##     vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##     main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##     CH20_between_1_and_2 + CH20_more_than_2 + monitor_cal + physical_act_1_2 +
##     physical_act_2_4 + tech_1_hour + tech_2_hours_or_more + alcohol_always +
##     alcohol_frequently + alcohol_sometimes + mtrans_automobile +
##     mtrans_bike + mtrans_public_transportation
##
##
##              Df Sum of Sq    RSS    AIC
## - CH20_between_1_and_2      1         26 305541 6993.6
## - alcohol_frequently         1         54 305569 6993.7
## - mtrans_bike                 1         58 305573 6993.7
## - alcohol_always             1        179 305694 6994.2
## - food_inbetween_always      1        381 305896 6995.0
## <none>                        305515 6995.4
## - tech_2_hours_or_more      1         548 306063 6995.7
## - mtrans_public_transportation 1         762 306277 6996.6
## - physical_act_2_4           1        1022 306537 6997.7
## - monitor_cal                1        1192 306707 6998.4
## - mtrans_automobile          1        1543 307058 6999.8
## - eat_caloric                1        1772 307287 7000.8
## - vegetables_sometimes       1        1935 307451 7001.4
## - Gender                    1        2917 308432 7005.5
## - tech_1_hour               1        3022 308537 7005.9
## - alcohol_sometimes          1        4822 310337 7013.3
## - main_meals_Btw_1_2        1        6938 312453 7021.9
## - CH20_more_than_2          1        7374 312889 7023.6
## - physical_act_1_2          1        8113 313628 7026.6
## - vegetables_always         1       13669 319185 7048.9
## - Age                       1       17150 322665 7062.6
## - family_hist               1       33606 339121 7125.6
## - food_inbetween_frequently 1       34095 339610 7127.4
## - main_meals_More_than_3     1       34346 339862 7128.3

```

```

## - Height                1      73963 379478 7267.9
##
## Step:  AIC=6993.56
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##      vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##      main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##      CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
##      tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_frequently +
##      alcohol_sometimes + mtrans_automobile + mtrans_bike + mtrans_public_transportation
##
##              Df Sum of Sq    RSS    AIC
## - mtrans_bike      1         55 305596 6991.8
## - alcohol_frequently 1         55 305596 6991.8
## - alcohol_always    1        184 305724 6992.3
## - food_inbetween_always 1        383 305924 6993.1
## <none>              305541 6993.6
## - tech_2_hours_or_more 1        557 306098 6993.9
## - mtrans_public_transportation 1        770 306311 6994.7
## - physical_act_2_4    1       1054 306595 6995.9
## - monitor_cal         1       1176 306717 6996.4
## - mtrans_automobile   1       1543 307084 6997.9
## - eat_caloric         1       1783 307324 6998.9
## - vegetables_sometimes 1       1952 307492 6999.6
## - Gender              1       2918 308458 7003.6
## - tech_1_hour         1       3031 308572 7004.1
## - alcohol_sometimes    1       4859 310400 7011.5
## - main_meals_Btw_1_2   1       6914 312455 7019.9
## - CH20_more_than_2     1       7968 313509 7024.1
## - physical_act_1_2     1       8091 313632 7024.6
## - vegetables_always    1      13723 319264 7047.2
## - Age                 1      17130 322671 7060.6
## - main_meals_More_than_3 1      34377 339917 7126.5
## - family_hist          1      34502 340042 7127.0
## - food_inbetween_frequently 1      35247 340787 7129.8
## - Height              1      74497 380038 7267.8
##
## Step:  AIC=6991.78
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##      vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##      main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##      CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
##      tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_frequently +
##      alcohol_sometimes + mtrans_automobile + mtrans_public_transportation
##
##              Df Sum of Sq    RSS    AIC
## - alcohol_frequently    1         53 305649 6990.0
## - alcohol_always        1        192 305787 6990.6
## - food_inbetween_always 1        397 305993 6991.4

```



```

## <none> 305596 6991.8
## - tech_2_hours_or_more 1 569 306164 6992.1
## - mtrans_public_transportation 1 947 306542 6993.7
## - physical_act_2_4 1 1064 306660 6994.2
## - monitor_cal 1 1193 306789 6994.7
## - mtrans_automobile 1 1495 307091 6996.0
## - eat_caloric 1 1774 307370 6997.1
## - vegetables_sometimes 1 1940 307536 6997.8
## - Gender 1 2966 308562 7002.0
## - tech_1_hour 1 3036 308632 7002.3
## - alcohol_sometimes 1 4874 310470 7009.8
## - main_meals_Btw_1_2 1 6874 312469 7017.9
## - CH20_more_than_2 1 7994 313590 7022.5
## - physical_act_1_2 1 8120 313716 7023.0
## - vegetables_always 1 13704 319299 7045.3
## - Age 1 17101 322697 7058.7
## - main_meals_More_than_3 1 34322 339918 7124.5
## - family_hist 1 34559 340154 7125.4
## - food_inbetween_frequently 1 35192 340787 7127.8
## - Height 1 74674 380270 7266.6
##
## Step: AIC=6990
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
## vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
## main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
## CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
## tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_sometimes +
## mtrans_automobile + mtrans_public_transportation
##
## Df Sum of Sq RSS AIC
## - alcohol_always 1 194 305843 6988.8
## - food_inbetween_always 1 414 306063 6989.7
## <none> 305649 6990.0
## - tech_2_hours_or_more 1 548 306197 6990.3
## - mtrans_public_transportation 1 935 306584 6991.9
## - physical_act_2_4 1 1077 306725 6992.5
## - monitor_cal 1 1230 306879 6993.1
## - mtrans_automobile 1 1519 307168 6994.3
## - eat_caloric 1 1766 307415 6995.3
## - vegetables_sometimes 1 1957 307606 6996.1
## - Gender 1 2968 308617 7000.2
## - tech_1_hour 1 3005 308654 7000.4
## - alcohol_sometimes 1 5478 311127 7010.5
## - main_meals_Btw_1_2 1 6846 312495 7016.0
## - CH20_more_than_2 1 7945 313594 7020.5
## - physical_act_1_2 1 8110 313759 7021.2
## - vegetables_always 1 13749 319398 7043.7
## - Age 1 17050 322699 7056.7

```

```

## - main_meals_More_than_3      1      34337 339986 7122.8
## - family_hist                 1      34597 340246 7123.8
## - food_inbetween_frequently   1      35440 341089 7126.9
## - Height                     1      74647 380296 7264.6
##
## Step:  AIC=6988.81
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##     vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##     main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##     CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
##     tech_1_hour + tech_2_hours_or_more + alcohol_sometimes +
##     mtrans_automobile + mtrans_public_transportation
##
##                                     Df Sum of Sq    RSS    AIC
## - food_inbetween_always           1         415 306258 6988.5
## <none>                             305843 6988.8
## - tech_2_hours_or_more            1         553 306396 6989.1
## - mtrans_public_transportation     1         834 306678 6990.3
## - physical_act_2_4                1        1049 306892 6991.1
## - monitor_cal                    1        1243 307087 6991.9
## - mtrans_automobile               1        1687 307530 6993.8
## - eat_caloric                    1        1826 307670 6994.3
## - vegetables_sometimes            1        1971 307814 6994.9
## - tech_1_hour                    1        2934 308778 6998.9
## - Gender                         1        2938 308782 6998.9
## - alcohol_sometimes               1        5449 311292 7009.2
## - main_meals_Btw_1_2              1        6764 312607 7014.5
## - CH20_more_than_2                1        7912 313755 7019.1
## - physical_act_1_2                1        8226 314070 7020.4
## - vegetables_always               1       13705 319549 7042.3
## - Age                            1       17016 322860 7055.4
## - main_meals_More_than_3          1       34420 340264 7121.8
## - family_hist                     1       34869 340713 7123.5
## - food_inbetween_frequently       1       35247 341090 7124.9
## - Height                         1       74523 380367 7262.9
##
## Step:  AIC=6988.52
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##     vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##     main_meals_More_than_3 + food_inbetween_frequently + CH20_more_than_2 +
##     monitor_cal + physical_act_1_2 + physical_act_2_4 + tech_1_hour +
##     tech_2_hours_or_more + alcohol_sometimes + mtrans_automobile +
##     mtrans_public_transportation
##
##                                     Df Sum of Sq    RSS    AIC
## <none>                             306258 6988.5
## - tech_2_hours_or_more            1         574 306832 6988.9
## - mtrans_public_transportation     1         926 307184 6990.3

```

```
## - physical_act_2_4      1      1012 307270 6990.7
## - monitor_cal           1      1335 307593 6992.0
## - mtrans_automobile     1      1602 307860 6993.1
## - eat_caloric           1      1883 308141 6994.3
## - vegetables_sometimes  1      2252 308510 6995.8
## - tech_1_hour           1      2952 309210 6998.7
## - Gender                1      3001 309259 6998.9
## - alcohol_sometimes     1      5572 311830 7009.3
## - main_meals_Btw_1_2    1      6579 312837 7013.4
## - CH20_more_than_2      1      8041 314299 7019.3
## - physical_act_1_2      1      8493 314751 7021.2
## - vegetables_always     1     13718 319976 7042.0
## - Age                   1     17193 323451 7055.7
## - food_inbetween_frequently 1    34859 341116 7123.0
## - main_meals_More_than_3 1    35162 341419 7124.1
## - family_hist           1    36034 342292 7127.4
## - Height                1    74621 380879 7262.6
```

```
summary(lm_backward_obesity)
```

```
##
## Call:
## lm(formula = Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##     vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##     main_meals_More_than_3 + food_inbetween_frequently + CH20_more_than_2 +
##     monitor_cal + physical_act_1_2 + physical_act_2_4 + tech_1_hour +
##     tech_2_hours_or_more + alcohol_sometimes + mtrans_automobile +
##     mtrans_public_transportation, data = train.set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -54.796  -9.651   0.799   9.613  53.566
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -157.25649    12.73878  -12.345 < 2e-16 ***
## Gender          4.32640     1.23821   3.494 0.000492 ***
## Age            0.81187     0.09707   8.363 < 2e-16 ***
## Height        121.98641     7.00108  17.424 < 2e-16 ***
## family_hist    15.62919     1.29081  12.108 < 2e-16 ***
## eat_caloric     4.07848     1.47335   2.768 0.005721 **
## vegetables_sometimes -10.25052     3.38653  -3.027 0.002522 **
## vegetables_always  -7.35775     0.98489  -7.471 1.50e-13 ***
## main_meals_Btw_1_2  -5.33322     1.03086  -5.174 2.67e-07 ***
## main_meals_More_than_3 -18.11019     1.51417 -11.960 < 2e-16 ***
## food_inbetween_frequently -17.48159     1.46795 -11.909 < 2e-16 ***
## CH20_more_than_2     5.42381     0.94825   5.720 1.33e-08 ***
```

```
## monitor_cal -5.18724 2.22576 -2.331 0.019936 *
## physical_act_1_2 -9.53018 1.62130 -5.878 5.32e-09 ***
## physical_act_2_4 2.01635 0.99354 2.029 0.042624 *
## tech_1_hour -5.36631 1.54856 -3.465 0.000548 ***
## tech_2_hours_or_more -1.51977 0.99466 -1.528 0.126783
## alcohol_sometimes 4.78278 1.00453 4.761 2.15e-06 ***
## mtrans_automobile -6.88857 2.69795 -2.553 0.010790 *
## mtrans_public_transportation 4.79038 2.46791 1.941 0.052475 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.68 on 1246 degrees of freedom
## Multiple R-squared: 0.6455, Adjusted R-squared: 0.6401
## F-statistic: 119.4 on 19 and 1246 DF, p-value: < 2.2e-16
```

```
# formula: Weight ~ Gender + Age + Height +
# family_hist + eat_caloric +
# vegetables_sometimes + vegetables_always +
# main_meals_Btw_1_2 + main_meals_More_than_3
# + food_inbetween_frequently +
# CH20_more_than_2 + monitor_cal +
# physical_act_1_2 + physical_act_2_4 +
# tech_1_hour + tech_2_hours_or_more +
# alcohol_sometimes + mtrans_automobile +
# mtrans_public_transportation
```

```
# AIC: 6988.52
```

```
# Both
```

```
lm_both_obesity <- step(lm_weight, direction = "both")
```

```
## Start: AIC=6999.41
```

```
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
## vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
## main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
## food_inbetween_sometimes + smoke + CH20_between_1_and_2 +
## CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
## tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_frequently +
## alcohol_sometimes + mtrans_automobile + mtrans_bike + mtrans_public_transportation
```

	Df	Sum of Sq	RSS	AIC
## - smoke	1	1	305507	6997.4
## - food_inbetween_sometimes	1	8	305514	6997.4
## - CH20_between_1_and_2	1	27	305533	6997.5
## - alcohol_frequently	1	51	305556	6997.6
## - mtrans_bike	1	59	305564	6997.7
## - food_inbetween_always	1	136	305642	6998.0

```

## - alcohol_always          1      179 305685 6998.2
## <none>                     305506 6999.4
## - tech_2_hours_or_more    1      528 306034 6999.6
## - mtrans_public_transportation 1      759 306265 7000.6
## - physical_act_2_4        1     1013 306518 7001.6
## - monitor_cal              1     1184 306689 7002.3
## - mtrans_automobile        1     1544 307049 7003.8
## - eat_caloric              1     1774 307280 7004.7
## - vegetables_sometimes     1     1883 307389 7005.2
## - Gender                   1     2872 308378 7009.3
## - tech_1_hour              1     3017 308523 7009.9
## - alcohol_sometimes        1     4831 310336 7017.3
## - food_inbetween_frequently 1     6227 311732 7023.0
## - main_meals_Btw_1_2       1     6900 312405 7025.7
## - CH20_more_than_2         1     7223 312728 7027.0
## - physical_act_1_2         1     8120 313626 7030.6
## - vegetables_always        1    13305 318811 7051.4
## - Age                      1    16637 322143 7064.5
## - family_hist              1    32630 338136 7125.9
## - main_meals_More_than_3    1    33997 339502 7131.0
## - Height                   1    71699 377204 7264.3
##
## Step:  AIC=6997.41
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##      vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##      main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##      food_inbetween_sometimes + CH20_between_1_and_2 + CH20_more_than_2 +
##      monitor_cal + physical_act_1_2 + physical_act_2_4 + tech_1_hour +
##      tech_2_hours_or_more + alcohol_always + alcohol_frequently +
##      alcohol_sometimes + mtrans_automobile + mtrans_bike + mtrans_public_transportation
##
##              Df Sum of Sq    RSS    AIC
## - food_inbetween_sometimes      1      8 305515 6995.4
## - CH20_between_1_and_2          1     27 305534 6995.5
## - alcohol_frequently             1     53 305560 6995.6
## - mtrans_bike                   1     58 305565 6995.7
## - food_inbetween_always          1    136 305643 6996.0
## - alcohol_always                 1    179 305686 6996.2
## <none>                           305507 6997.4
## - tech_2_hours_or_more          1     527 306034 6997.6
## - mtrans_public_transportation    1     760 306267 6998.6
## + smoke                         1        1 305506 6999.4
## - physical_act_2_4              1    1012 306519 6999.6
## - monitor_cal                   1    1194 306701 7000.4
## - mtrans_automobile              1    1547 307054 7001.8
## - eat_caloric                   1    1776 307283 7002.8
## - vegetables_sometimes           1    1884 307391 7003.2
## - Gender                        1    2871 308378 7007.3

```

```

## - tech_1_hour          1      3030 308537 7007.9
## - alcohol_sometimes    1      4830 310336 7015.3
## - food_inbetween_frequently 1      6234 311741 7021.0
## - main_meals_Btw_1_2   1      6901 312408 7023.7
## - CH20_more_than_2     1      7315 312822 7025.4
## - physical_act_1_2     1      8119 313626 7028.6
## - vegetables_always    1     13310 318817 7049.4
## - Age                  1     17080 322587 7064.3
## - family_hist          1     32631 338138 7123.9
## - main_meals_More_than_3 1     34024 339531 7129.1
## - Height               1     72330 377837 7264.4
##
## Step:  AIC=6995.45
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##      vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##      main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##      CH20_between_1_and_2 + CH20_more_than_2 + monitor_cal + physical_act_1_2 +
##      physical_act_2_4 + tech_1_hour + tech_2_hours_or_more + alcohol_always +
##      alcohol_frequently + alcohol_sometimes + mtrans_automobile +
##      mtrans_bike + mtrans_public_transportation
##
##
##              Df Sum of Sq    RSS    AIC
## - CH20_between_1_and_2      1      26 305541 6993.6
## - alcohol_frequently        1      54 305569 6993.7
## - mtrans_bike                1      58 305573 6993.7
## - alcohol_always            1     179 305694 6994.2
## - food_inbetween_always     1     381 305896 6995.0
## <none>                      305515 6995.4
## - tech_2_hours_or_more      1     548 306063 6995.7
## - mtrans_public_transportation 1     762 306277 6996.6
## + food_inbetween_sometimes   1       8 305507 6997.4
## + smoke                      1       2 305514 6997.4
## - physical_act_2_4          1    1022 306537 6997.7
## - monitor_cal               1    1192 306707 6998.4
## - mtrans_automobile         1    1543 307058 6999.8
## - eat_caloric               1    1772 307287 7000.8
## - vegetables_sometimes      1    1935 307451 7001.4
## - Gender                   1    2917 308432 7005.5
## - tech_1_hour              1    3022 308537 7005.9
## - alcohol_sometimes         1    4822 310337 7013.3
## - main_meals_Btw_1_2       1    6938 312453 7021.9
## - CH20_more_than_2         1    7374 312889 7023.6
## - physical_act_1_2         1    8113 313628 7026.6
## - vegetables_always        1   13669 319185 7048.9
## - Age                      1   17150 322665 7062.6
## - family_hist              1   33606 339121 7125.6
## - food_inbetween_frequently 1   34095 339610 7127.4
## - main_meals_More_than_3    1   34346 339862 7128.3

```

```

## - Height          1      73963 379478 7267.9
##
## Step:  AIC=6993.56
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##      vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##      main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##      CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
##      tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_frequently +
##      alcohol_sometimes + mtrans_automobile + mtrans_bike + mtrans_public_transportation
##
##              Df Sum of Sq    RSS    AIC
## - mtrans_bike      1         55 305596 6991.8
## - alcohol_frequently 1         55 305596 6991.8
## - alcohol_always    1        184 305724 6992.3
## - food_inbetween_always 1        383 305924 6993.1
## <none>              305541 6993.6
## - tech_2_hours_or_more 1        557 306098 6993.9
## - mtrans_public_transportation 1        770 306311 6994.7
## + CH20_between_1_and_2 1         26 305515 6995.4
## + food_inbetween_sometimes 1          6 305534 6995.5
## + smoke            1          2 305539 6995.5
## - physical_act_2_4  1       1054 306595 6995.9
## - monitor_cal      1       1176 306717 6996.4
## - mtrans_automobile 1       1543 307084 6997.9
## - eat_caloric      1       1783 307324 6998.9
## - vegetables_sometimes 1       1952 307492 6999.6
## - Gender           1       2918 308458 7003.6
## - tech_1_hour      1       3031 308572 7004.1
## - alcohol_sometimes 1       4859 310400 7011.5
## - main_meals_Btw_1_2 1       6914 312455 7019.9
## - CH20_more_than_2  1       7968 313509 7024.1
## - physical_act_1_2  1       8091 313632 7024.6
## - vegetables_always 1      13723 319264 7047.2
## - Age              1      17130 322671 7060.6
## - main_meals_More_than_3 1      34377 339917 7126.5
## - family_hist      1      34502 340042 7127.0
## - food_inbetween_frequently 1      35247 340787 7129.8
## - Height           1      74497 380038 7267.8
##
## Step:  AIC=6991.78
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##      vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##      main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##      CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
##      tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_frequently +
##      alcohol_sometimes + mtrans_automobile + mtrans_public_transportation
##
##              Df Sum of Sq    RSS    AIC

```

```

## - alcohol_frequently      1      53 305649 6990.0
## - alcohol_always          1     192 305787 6990.6
## - food_inbetween_always   1     397 305993 6991.4
## <none>                     305596 6991.8
## - tech_2_hours_or_more    1     569 306164 6992.1
## + mtrans_bike              1      55 305541 6993.6
## + CH20_between_1_and_2     1      23 305573 6993.7
## - mtrans_public_transportation 1     947 306542 6993.7
## + food_inbetween_sometimes 1       6 305590 6993.8
## + smoke                    1       2 305594 6993.8
## - physical_act_2_4         1    1064 306660 6994.2
## - monitor_cal              1    1193 306789 6994.7
## - mtrans_automobile        1    1495 307091 6996.0
## - eat_caloric              1    1774 307370 6997.1
## - vegetables_sometimes     1    1940 307536 6997.8
## - Gender                   1    2966 308562 7002.0
## - tech_1_hour              1    3036 308632 7002.3
## - alcohol_sometimes        1    4874 310470 7009.8
## - main_meals_Btw_1_2       1    6874 312469 7017.9
## - CH20_more_than_2         1    7994 313590 7022.5
## - physical_act_1_2         1    8120 313716 7023.0
## - vegetables_always        1   13704 319299 7045.3
## - Age                      1   17101 322697 7058.7
## - main_meals_More_than_3    1   34322 339918 7124.5
## - family_hist              1   34559 340154 7125.4
## - food_inbetween_frequently 1   35192 340787 7127.8
## - Height                   1   74674 380270 7266.6
##
## Step: AIC=6990
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##      vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##      main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
##      CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
##      tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_sometimes +
##      mtrans_automobile + mtrans_public_transportation
##
##
##      Df Sum of Sq    RSS    AIC
## - alcohol_always      1      194 305843 6988.8
## - food_inbetween_always 1      414 306063 6989.7
## <none>                  305649 6990.0
## - tech_2_hours_or_more 1      548 306197 6990.3
## + alcohol_frequently    1       53 305596 6991.8
## + mtrans_bike           1       53 305596 6991.8
## - mtrans_public_transportation 1     935 306584 6991.9
## + CH20_between_1_and_2  1       24 305625 6991.9
## + food_inbetween_sometimes 1        7 305642 6992.0
## + smoke                 1        4 305645 6992.0
## - physical_act_2_4      1    1077 306725 6992.5

```



```

## - monitor_cal 1 1230 306879 6993.1
## - mtrans_automobile 1 1519 307168 6994.3
## - eat_caloric 1 1766 307415 6995.3
## - vegetables_sometimes 1 1957 307606 6996.1
## - Gender 1 2968 308617 7000.2
## - tech_1_hour 1 3005 308654 7000.4
## - alcohol_sometimes 1 5478 311127 7010.5
## - main_meals_Btw_1_2 1 6846 312495 7016.0
## - CH20_more_than_2 1 7945 313594 7020.5
## - physical_act_1_2 1 8110 313759 7021.2
## - vegetables_always 1 13749 319398 7043.7
## - Age 1 17050 322699 7056.7
## - main_meals_More_than_3 1 34337 339986 7122.8
## - family_hist 1 34597 340246 7123.8
## - food_inbetween_frequently 1 35440 341089 7126.9
## - Height 1 74647 380296 7264.6
##
## Step: AIC=6988.81
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
## vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
## main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently +
## CH20_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 +
## tech_1_hour + tech_2_hours_or_more + alcohol_sometimes +
## mtrans_automobile + mtrans_public_transportation
##
## Df Sum of Sq RSS AIC
## - food_inbetween_always 1 415 306258 6988.5
## <none> 305843 6988.8
## - tech_2_hours_or_more 1 553 306396 6989.1
## + alcohol_always 1 194 305649 6990.0
## - mtrans_public_transportation 1 834 306678 6990.3
## + mtrans_bike 1 62 305782 6990.6
## + alcohol_frequently 1 56 305787 6990.6
## + CH20_between_1_and_2 1 28 305815 6990.7
## + food_inbetween_sometimes 1 7 305837 6990.8
## + smoke 1 4 305839 6990.8
## - physical_act_2_4 1 1049 306892 6991.1
## - monitor_cal 1 1243 307087 6991.9
## - mtrans_automobile 1 1687 307530 6993.8
## - eat_caloric 1 1826 307670 6994.3
## - vegetables_sometimes 1 1971 307814 6994.9
## - tech_1_hour 1 2934 308778 6998.9
## - Gender 1 2938 308782 6998.9
## - alcohol_sometimes 1 5449 311292 7009.2
## - main_meals_Btw_1_2 1 6764 312607 7014.5
## - CH20_more_than_2 1 7912 313755 7019.1
## - physical_act_1_2 1 8226 314070 7020.4
## - vegetables_always 1 13705 319549 7042.3

```

```
## - Age 1 17016 322860 7055.4
## - main_meals_More_than_3 1 34420 340264 7121.8
## - family_hist 1 34869 340713 7123.5
## - food_inbetween_frequently 1 35247 341090 7124.9
## - Height 1 74523 380367 7262.9
##
## Step: AIC=6988.52
## Weight ~ Gender + Age + Height + family_hist + eat_caloric +
## vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
## main_meals_More_than_3 + food_inbetween_frequently + CH20_more_than_2 +
## monitor_cal + physical_act_1_2 + physical_act_2_4 + tech_1_hour +
## tech_2_hours_or_more + alcohol_sometimes + mtrans_automobile +
## mtrans_public_transportation
##
## Df Sum of Sq RSS AIC
## <none> 306258 6988.5
## + food_inbetween_always 1 415 305843 6988.8
## - tech_2_hours_or_more 1 574 306832 6988.9
## + food_inbetween_sometimes 1 266 305992 6989.4
## + alcohol_always 1 195 306063 6989.7
## + mtrans_bike 1 77 306181 6990.2
## + alcohol_frequently 1 73 306185 6990.2
## - mtrans_public_transportation 1 926 307184 6990.3
## + CH20_between_1_and_2 1 30 306228 6990.4
## + smoke 1 9 306249 6990.5
## - physical_act_2_4 1 1012 307270 6990.7
## - monitor_cal 1 1335 307593 6992.0
## - mtrans_automobile 1 1602 307860 6993.1
## - eat_caloric 1 1883 308141 6994.3
## - vegetables_sometimes 1 2252 308510 6995.8
## - tech_1_hour 1 2952 309210 6998.7
## - Gender 1 3001 309259 6998.9
## - alcohol_sometimes 1 5572 311830 7009.3
## - main_meals_Btw_1_2 1 6579 312837 7013.4
## - CH20_more_than_2 1 8041 314299 7019.3
## - physical_act_1_2 1 8493 314751 7021.2
## - vegetables_always 1 13718 319976 7042.0
## - Age 1 17193 323451 7055.7
## - food_inbetween_frequently 1 34859 341116 7123.0
## - main_meals_More_than_3 1 35162 341419 7124.1
## - family_hist 1 36034 342292 7127.4
## - Height 1 74621 380879 7262.6
```

```
summary(lm_both_obesity)
```

```
##
## Call:
```

```
## lm(formula = Weight ~ Gender + Age + Height + family_hist + eat_caloric +
##     vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 +
##     main_meals_More_than_3 + food_inbetween_frequently + CH20_more_than_2 +
##     monitor_cal + physical_act_1_2 + physical_act_2_4 + tech_1_hour +
##     tech_2_hours_or_more + alcohol_sometimes + mtrans_automobile +
##     mtrans_public_transportation, data = train.set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -54.796  -9.651   0.799   9.613  53.566
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -157.25649    12.73878  -12.345 < 2e-16 ***
## Gender              4.32640     1.23821   3.494 0.000492 ***
## Age                0.81187     0.09707   8.363 < 2e-16 ***
## Height           121.98641     7.00108  17.424 < 2e-16 ***
## family_hist       15.62919     1.29081  12.108 < 2e-16 ***
## eat_caloric        4.07848     1.47335   2.768 0.005721 **
## vegetables_sometimes -10.25052     3.38653  -3.027 0.002522 **
## vegetables_always   -7.35775     0.98489  -7.471 1.50e-13 ***
## main_meals_Btw_1_2   -5.33322     1.03086  -5.174 2.67e-07 ***
## main_meals_More_than_3 -18.11019     1.51417 -11.960 < 2e-16 ***
## food_inbetween_frequently -17.48159     1.46795 -11.909 < 2e-16 ***
## CH20_more_than_2      5.42381     0.94825   5.720 1.33e-08 ***
## monitor_cal        -5.18724     2.22576  -2.331 0.019936 *
## physical_act_1_2    -9.53018     1.62130  -5.878 5.32e-09 ***
## physical_act_2_4      2.01635     0.99354   2.029 0.042624 *
## tech_1_hour        -5.36631     1.54856  -3.465 0.000548 ***
## tech_2_hours_or_more -1.51977     0.99466  -1.528 0.126783
## alcohol_sometimes     4.78278     1.00453   4.761 2.15e-06 ***
## mtrans_automobile    -6.88857     2.69795  -2.553 0.010790 *
## mtrans_public_transportation  4.79038     2.46791   1.941 0.052475 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.68 on 1246 degrees of freedom
## Multiple R-squared:  0.6455, Adjusted R-squared:  0.6401
## F-statistic: 119.4 on 19 and 1246 DF, p-value: < 2.2e-16
```

```
# AIC: 6988.52 model:Weight ~ Gender + Age +
# Height + family_hist + eat_caloric +
# vegetables_sometimes + vegetables_always +
# main_meals_Btw_1_2 + main_meals_More_than_3
# + food_inbetween_frequently +
# CH20_more_than_2 + monitor_cal +
# physical_act_1_2 + physical_act_2_4 +
# tech_1_hour + tech_2_hours_or_more +
```

```
# alcohol_sometimes + mtrans_automobile +
# mtrans_public_transportation

# Simplet model(both, backward)

# Add comments +Assumptions
```

For the forward model, the stepwise selection shows us that the best model is:

Weight ~ Gender + Age + Height + family_hist + eat_caloric + vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 + main_meals_More_than_3 + food_inbetween_always + food_inbetween_frequently + food_inbetween_sometimes + smoke + CH2O_between_1_and_2 + CH2O_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 + tech_1_hour + tech_2_hours_or_more + alcohol_always + alcohol_frequently + alcohol_sometimes + mtrans_automobile + mtrans_bike + mtrans_public_transportation

This is in fact the same model as the full model. It has an AIC of 6999.41, an R-Squared of 0.6464 and an adjusted R-Squared of 0.639.

For the backward model, the stepwise selection shows us that the best model is:

Weight ~ Gender + Age + Height + family_hist + eat_caloric + vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 + main_meals_More_than_3 + food_inbetween_frequently + CH2O_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 + tech_1_hour + tech_2_hours_or_more + alcohol_sometimes + mtrans_automobile + mtrans_public_transportation

This model is a reduced version of the full model. The AIC is 6988.52, the R-Squared is 0.6455 and the adjusted R-Squared is 0.6401.

For the both model we obtain the same results as the backward model. The best model is:

Weight ~ Gender + Age + Height + family_hist + eat_caloric + vegetables_sometimes + vegetables_always + main_meals_Btw_1_2 + main_meals_More_than_3 + food_inbetween_frequently + CH2O_more_than_2 + monitor_cal + physical_act_1_2 + physical_act_2_4 + tech_1_hour + tech_2_hours_or_more + alcohol_sometimes + mtrans_automobile + mtrans_public_transportation

This model is a reduced version of the full model. The AIC is 6988.52, the R-Squared is 0.6455 and the adjusted R-Squared is 0.6401.

When looking at all three models, the best model would seem to be the backward model (or the both model). Its adjusted R-Squared is higher than the forward model by very little but is reduced and therefore favorable. We have very very similar results and insights from all

three models but the backward model and the both model allow us to obtain those insights without having to drag around those variables that are not significant.

To confirm our choice of model for the linear regression, we will proceed with the validation of the accuracy of the predictions on the validation set with the help of 3 metrics: RMSE, Mean error and MAPE.

```
# Predictions on the validation set

# Forward model:
forward_pred_obesity <- predict(lm_forward_obesity,
                                valid.set)

# RMSE
gofRMSE(valid.set$Weight, forward_pred_obesity,
         dgt = 3) #16.376
```

```
## [1] 16.376
```

```
# Mean error
gofME(valid.set$Weight, forward_pred_obesity,
       dgt = 3) #1.038
```

```
## [1] 1.038
```

```
# MAPE
gofMAPE(valid.set$Weight, forward_pred_obesity,
         dgt = 3) #16.344
```

```
## [1] 16.344
```

```
# Backward model:
backward_pred_obesity <- predict(lm_backward_obesity,
                                  valid.set)

# RMSE
gofRMSE(valid.set$Weight, backward_pred_obesity,
         dgt = 3) #16.416
```

```
## [1] 16.416
```

```
# Mean error
gofME(valid.set$Weight, backward_pred_obesity,
       dgt = 3) #1.002
```

```
## [1] 1.002
```

```
# MAPE
gofMAPE(valid.set$Weight, backward_pred_obesity,
        dgt = 3) #16.363
```

```
## [1] 16.363
```

```
# Both model:
both_pred_obesity <- predict(lm_both_obesity,
                             valid.set)
```

```
# RMSE
gofRMSE(valid.set$Weight, both_pred_obesity, dgt = 3) #16.416
```

```
## [1] 16.416
```

```
# Mean error
gofME(valid.set$Weight, both_pred_obesity, dgt = 3) #1.002
```

```
## [1] 1.002
```

```
# MAPE
gofMAPE(valid.set$Weight, both_pred_obesity, dgt = 3) #16.363
```

```
## [1] 16.363
```

```
# Add comments +Assumptions
```

Just as we had mentionned above, the backward model and the both model seem to represent the best model for our data. The difference in the three metrics for each model are very very small. This enables us to choose the backward/both model as the best model, since it yields very similar results as the full model, without all the cumbersome variables that are not relevant in the full (forward) model.

k-Nearest Neighbors

First of all, we should normalize the data, since we have different scales and big values can dominate small values! We normalize only the numerical data.

```

# Normalizing the data :

normalize <- function(x) {
  return((x - min(x))/(max(x) - min(x)))
}

train.set.norm <- as.data.frame(lapply(train.set[,
  c(2:4)], normalize))

valid.set.norm <- as.data.frame(lapply(valid.set[,
  c(2:4)], normalize))

# Regrouping into final dataset, with
# replacement of non-normalized variables :

train.final <- cbind(train.set.norm, train.set[,
  c(1, 5:36)])

valid.final <- cbind(valid.set.norm, valid.set[,
  c(1, 5:36)])

```

A “denormalize” function will be very handy when converting the predicted weight back to its normal scale :

```

denormalize <- function(x, y) {

  return((x * (max(y) - min(y))) + min(y))

}

```

Now we will run the model JUST in order to identify different values of the **Root Mean Squared Error (RMSE)** :

```

# Creating dataframe :

rmse.df = data.frame(k = seq(1, 40, 1), RMSE = rep(0,
  40))

# Running the model :

set.seed(1)

for (i in 1:40) {

  knn.pred = knn(train = train.final[, -3],

```

```

        test = valid.final[, -3], cl = train.final[,
            3], k = i)

k_nn = as.numeric(as.character(knn.pred))

predicted = denormalize(k_nn, valid.set[,
    4])

rmse.df[i, 2] = sqrt(mean((predicted - valid.set[,
    4])^2))
}

# Plotting results :

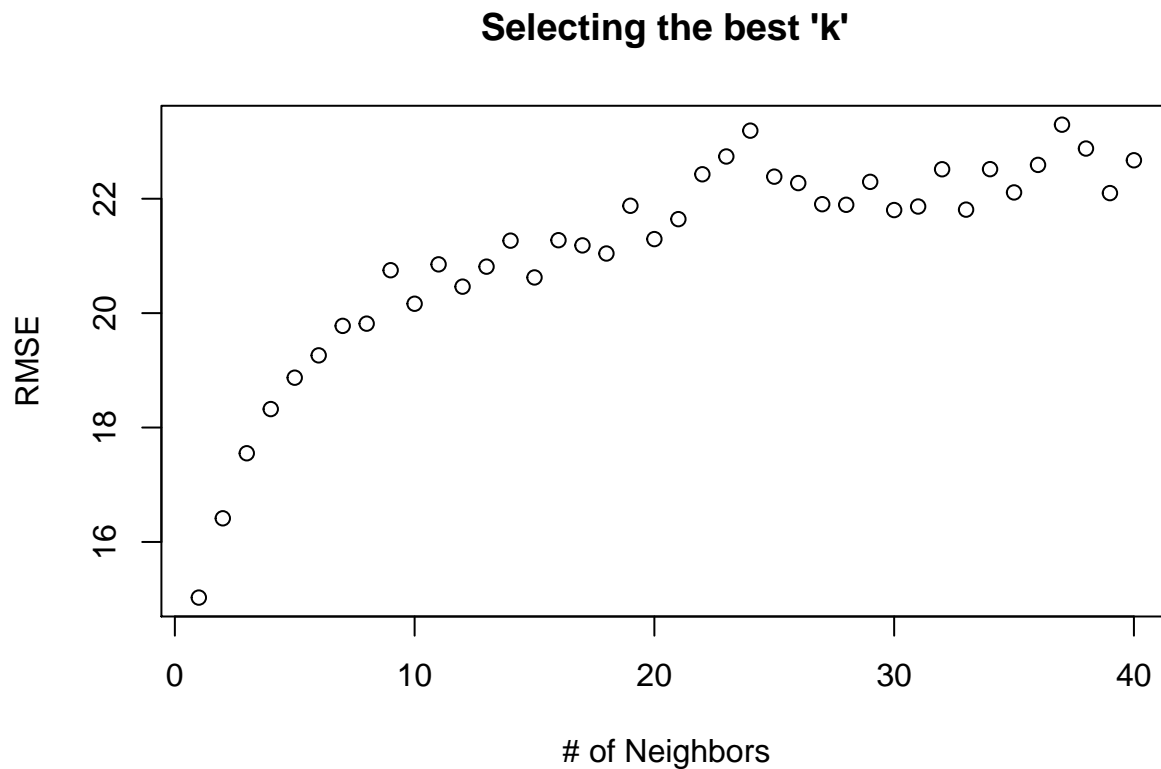
pander(rmse.df)

```

k	RMSE
1	15.03
2	16.41
3	17.55
4	18.32
5	18.87
6	19.26
7	19.78
8	19.82
9	20.75
10	20.16
11	20.85
12	20.46
13	20.81
14	21.27
15	20.62
16	21.27
17	21.18
18	21.04
19	21.88
20	21.29
21	21.64
22	22.43
23	22.74
24	23.19
25	22.39
26	22.27
27	21.9
28	21.89
29	22.3

k	RMSE
30	21.8
31	21.86
32	22.52
33	21.81
34	22.52
35	22.11
36	22.59
37	23.29
38	22.88
39	22.1
40	22.67

```
plot(rmse.df$k, rmse.df$RMSE, xlab = "# of Neighbors",
     ylab = "RMSE", main = "Selecting the best 'k'")
```



We can see from this plot (but also from the table above) that the best k is equal to 1. With this $k = 1$, we are able to minimize the RMSE.

Running the FINAL model (with $k = 1$):

```
set.seed(1)
```

```

k_nn <- knn(train = train.final[, -3], test = valid.final[,
  -3], cl = train.final[, 3], k = 1)

k_nn = as.numeric(as.character(k_nn))

predicted = denormalize(k_nn, valid.set[, 4])

RMSE = sqrt(mean((predicted - valid.set[, 4])^2))

RMSE

```

```
## [1] 15.02754
```

The resulting RMSE is equal to 15.02754.

Let's now do a Regression Tree and ultimately compare its RMSE with the one of the k-NN! Weird! If we change the set.seed, the results for RMSE change!! So, which set.seed to choose?

Regression Tree

We will first focus on selecting the appropriate value for the Complexity Parameter (CP hereafter), which is a “penalty” factor concerning the size of the tree. A smaller CP will result in a bigger tree, and viceversa.

To do this, we will do a Cross-Validation approach. The computer will create many different partitions of the dataset into training and validation, and we want to find the CP that corresponds to the minimum Cross-Validation error.

This procedure is meant to help addressing the **tree instability** issue.

```

# First run a quite big tree (CP = 0.00001) :

set.seed(1)

tree_1 <- rpart(Weight ~ ., data = train.set,
  method = "anova", control = rpart.control(cp = 1e-05,
    minbucket = 1, maxdepth = 10))

# We do a CV : must locate in the table the
# point from which the CV error starts to rise
# :

pander(printcp(tree_1))

```

Regression tree: `rpart(formula = Weight ~ ., data = train.set, method = "anova", control = rpart.control(cp = 1e-05, minbucket = 1, maxdepth = 10))`

Variables actually used in tree construction: [1] Age alcohol_frequently

[3] alcohol_no alcohol_sometimes

[5] CH2O_between_1_and_2 CH2O_less_than_a_liter

[7] CH2O_more_than_2 eat_caloric

[9] family_hist food_inbetween_always

[11] food_inbetween_frequently food_inbetween_no

[13] food_inbetween_sometimes Gender

[15] Height main_meals_Btw_1_2

[17] main_meals_More_than_3 main_meals_three

[19] mtrans_automobile mtrans_public_transportation [21] mtrans_walking physical_act_1_2

[23] physical_act_2_4 physical_act_do_not_have

[25] smoke tech_0_hours

[27] tech_1_hour tech_2_hours_or_more

[29] vegetables_always vegetables_never

[31] vegetables_sometimes

Root node error: $863970/1266 = 682.44$

n= 1266

	CP	nsplit	rel error	xerror	xstd
1	2.3455e-01	0	1.000000	1.00155	0.031612
2	0.645126	0.69143	0.024380	4.8885e-02	3
3	0.504555	0.53189	0.020062	6 3.9913e-02	5
4	0.45833	0.018216	8 3.0705e-02	7 0.384869	0.42132
5	0.017666	10 2.6283e-02	9 0.324267	0.39900	0.017759
6	0.016298	12 1.4922e-02	11 0.276595	0.31477	0.016084
7	0.015334	14 1.0733e-02	14 0.233073	0.27831	0.014811
8	0.014367	16 6.7285e-03	16 0.215608	0.25555	0.014318
9	0.014315	18 5.2398e-03	18 0.202454	0.24153	0.013774
10	0.013701	20 5.0756e-03	21 0.186902	0.23511	0.013704
11	0.013738	22 4.9464e-03	23 0.176773	0.23230	0.013641
12	0.013587	24 4.6777e-03	25 0.167013	0.22768	0.013597
13	0.013596	26 4.3459e-03	27 0.157790	0.22790	0.013646
14	0.013330	28 3.2234e-03	29 0.149442	0.20174	0.012413
15	0.012444	30 2.9916e-03	31 0.143126	0.19541	0.012417
16	0.012535	32 2.8075e-03	33 0.137276	0.19605	0.012634
17	0.012459	34 2.3938e-03	36 0.129177	0.19138	0.012666
18	0.014034	36 2.2876e-03	39 0.122025	0.19559	0.014017
19	0.014878	38 2.0813e-03	41 0.117554	0.20200	0.015113
20	0.015226	40 2.0205e-03	44 0.111362	0.20314	0.015217
21	0.015190	42 1.6986e-03	48 0.103567	0.19405	0.014685
22	0.014878	44 1.4059e-03	52 0.096878	0.19288	0.014810
23	0.015014	46 1.3517e-03	54 0.094071	0.19258	0.015012
24					47 1.3113e-03
25					55 0.092719
26					0.19280

0.015044 48 1.1903e-03 56 0.091408 0.19241 0.015191 49 1.1791e-03 57 0.090218 0.19372
 0.015187 50 1.0896e-03 58 0.089039 0.19450 0.015269 51 1.0834e-03 59 0.087949 0.19447
 0.015282 52 1.0806e-03 60 0.086866 0.19455 0.015281 53 1.0409e-03 61 0.085785 0.19446
 0.015286 54 1.0146e-03 62 0.084744 0.19419 0.015298 55 1.0047e-03 63 0.083730 0.19496
 0.015378 56 1.0036e-03 64 0.082725 0.19493 0.015378 57 9.9494e-04 65 0.081721 0.19493
 0.015378 58 9.4526e-04 66 0.080726 0.19454 0.015409 59 9.3369e-04 67 0.079781 0.19457
 0.015410 60 8.9659e-04 68 0.078847 0.19605 0.015490 61 8.8696e-04 69 0.077951 0.19705
 0.015511 62 8.7917e-04 70 0.077064 0.19663 0.015498 63 8.7664e-04 71 0.076185 0.19667
 0.015498 64 8.6872e-04 72 0.075308 0.19651 0.015504 65 8.5579e-04 73 0.074439 0.19666
 0.015512 66 7.9067e-04 74 0.073584 0.19647 0.015546 67 7.6136e-04 75 0.072793 0.19610
 0.015630 68 7.5874e-04 76 0.072032 0.19637 0.015648 69 7.3475e-04 77 0.071273 0.19719
 0.015678 70 7.1505e-04 78 0.070538 0.19624 0.015649 71 7.0739e-04 79 0.069823 0.19622
 0.015632 72 7.0003e-04 80 0.069116 0.19478 0.015445 73 6.9885e-04 81 0.068416 0.19442
 0.015445 74 6.9317e-04 82 0.067717 0.19345 0.015439 75 6.8168e-04 86 0.064944 0.19321
 0.015441 76 6.6291e-04 87 0.064262 0.19329 0.015462 77 6.4968e-04 88 0.063600 0.19127
 0.015334 78 6.1187e-04 90 0.062300 0.19127 0.015325 79 5.7700e-04 92 0.061076 0.19191
 0.015362 80 5.6197e-04 93 0.060499 0.19226 0.015343 81 5.6012e-04 94 0.059937 0.19295
 0.015399 82 5.5551e-04 95 0.059377 0.19295 0.015399 83 5.3345e-04 96 0.058822 0.19193
 0.015366 84 5.2949e-04 97 0.058288 0.19083 0.015368 85 5.2886e-04 98 0.057759 0.19080
 0.015368 86 5.2750e-04 99 0.057230 0.19089 0.015368 87 5.1885e-04 100 0.056703 0.19050
 0.015368 88 5.1087e-04 101 0.056184 0.19026 0.015362 89 4.9588e-04 103 0.055162 0.18986
 0.015354 90 4.8486e-04 104 0.054666 0.19013 0.015457 91 4.7963e-04 107 0.053211 0.19138
 0.015509 92 4.7509e-04 109 0.052252 0.19157 0.015508 93 4.5473e-04 110 0.051777 0.19067
 0.015507 94 4.5147e-04 111 0.051322 0.19007 0.015501 95 4.1846e-04 113 0.050419 0.19114
 0.015572 96 4.1701e-04 114 0.050001 0.19072 0.015578 97 4.0408e-04 115 0.049584 0.19061
 0.015585 98 3.8493e-04 117 0.048776 0.18972 0.015600 99 3.7460e-04 118 0.048391 0.18979
 0.015638 100 3.1367e-04 119 0.048016 0.18852 0.015599 101 3.0365e-04 122 0.047075 0.18864
 0.015630 102 3.0357e-04 123 0.046772 0.18918 0.015645 103 3.0112e-04 124 0.046468 0.18904
 0.015644 104 2.9037e-04 125 0.046167 0.18888 0.015646 105 2.8594e-04 126 0.045877 0.18885
 0.015646 106 2.8317e-04 128 0.045305 0.18871 0.015648 107 2.7999e-04 129 0.045022 0.18871
 0.015648 108 2.6409e-04 130 0.044742 0.18896 0.015658 109 2.6396e-04 131 0.044477 0.18867
 0.015658 110 2.5313e-04 132 0.044213 0.18853 0.015660 111 2.4891e-04 133 0.043960 0.18866
 0.015658 112 2.4321e-04 134 0.043711 0.18871 0.015661 113 2.2199e-04 135 0.043468 0.18916
 0.015678 114 2.1902e-04 136 0.043246 0.18922 0.015690 115 2.1789e-04 137 0.043027 0.19004
 0.015712 116 2.1609e-04 138 0.042809 0.19016 0.015713 117 2.0894e-04 139 0.042593 0.19264
 0.015872 118 2.0355e-04 140 0.042384 0.19292 0.015887 119 2.0283e-04 141 0.042181 0.19331
 0.015891 120 2.0174e-04 143 0.041775 0.19323 0.015892 121 1.9844e-04 144 0.041573 0.19336
 0.015891 122 1.9533e-04 145 0.041375 0.19338 0.015895 123 1.9267e-04 146 0.041180 0.19335
 0.015895 124 1.8506e-04 147 0.040987 0.19336 0.015896 125 1.7948e-04 149 0.040617 0.19340
 0.015924 126 1.7349e-04 150 0.040437 0.19324 0.015923 127 1.7314e-04 151 0.040264 0.19322
 0.015923 128 1.7105e-04 152 0.040091 0.19331 0.015925 129 1.6556e-04 153 0.039920 0.19332
 0.015925 130 1.6526e-04 154 0.039754 0.19325 0.015925 131 1.6449e-04 155 0.039589 0.19325
 0.015925 132 1.6278e-04 156 0.039424 0.19317 0.015926 133 1.6224e-04 157 0.039262 0.19342
 0.015943 134 1.6005e-04 158 0.039099 0.19356 0.015951 135 1.5929e-04 160 0.038779 0.19349
 0.015951 136 1.4593e-04 161 0.038620 0.19395 0.015975 137 1.4356e-04 162 0.038474 0.19491
 0.016003 138 1.4168e-04 163 0.038330 0.19491 0.016003 139 1.3681e-04 164 0.038189 0.19514
 0.016001 140 1.3370e-04 165 0.038052 0.19491 0.015997 141 1.3356e-04 166 0.037918 0.19483
 0.015969 142 1.3092e-04 167 0.037785 0.19494 0.015969 143 1.2761e-04 168 0.037654 0.19463

0.015969 144 1.2457e-04 170 0.037399 0.19460 0.015969 145 1.2415e-04 171 0.037274 0.19461
 0.015969 146 1.2226e-04 172 0.037150 0.19460 0.015969 147 1.1984e-04 173 0.037028 0.19416
 0.015970 148 1.1818e-04 174 0.036908 0.19435 0.015969 149 1.1353e-04 175 0.036790 0.19450
 0.016003 150 1.1202e-04 176 0.036676 0.19443 0.016105 151 1.1111e-04 177 0.036564 0.19458
 0.016105 152 1.1067e-04 178 0.036453 0.19470 0.016110 153 1.0856e-04 179 0.036342 0.19469
 0.016110 154 1.0700e-04 180 0.036234 0.19468 0.016110 155 1.0669e-04 181 0.036127 0.19468
 0.016110 156 1.0508e-04 182 0.036020 0.19479 0.016116 157 1.0460e-04 183 0.035915 0.19477
 0.016116 158 1.0419e-04 185 0.035706 0.19477 0.016116 159 1.0363e-04 186 0.035601 0.19476
 0.016116 160 9.9800e-05 187 0.035498 0.19471 0.016117 161 9.9712e-05 188 0.035398 0.19461
 0.016117 162 9.8810e-05 189 0.035298 0.19461 0.016117 163 9.3753e-05 190 0.035200 0.19485
 0.016136 164 9.2613e-05 191 0.035106 0.19516 0.016135 165 9.0585e-05 192 0.035013 0.19517
 0.016135 166 7.8345e-05 193 0.034923 0.19551 0.016170 167 7.7127e-05 194 0.034844 0.19558
 0.016194 168 7.5068e-05 195 0.034767 0.19512 0.016138 169 7.1679e-05 196 0.034692 0.19509
 0.016137 170 7.1332e-05 197 0.034620 0.19489 0.016124 171 7.0450e-05 198 0.034549 0.19496
 0.016125 172 6.7110e-05 199 0.034479 0.19495 0.016125 173 6.4467e-05 200 0.034411 0.19497
 0.016124 174 5.9261e-05 201 0.034347 0.19492 0.016125 175 5.9063e-05 202 0.034288 0.19487
 0.016123 176 5.7612e-05 203 0.034229 0.19501 0.016124 177 5.7072e-05 204 0.034171 0.19490
 0.016126 178 5.7045e-05 205 0.034114 0.19490 0.016126 179 5.7025e-05 206 0.034057 0.19490
 0.016126 180 5.5750e-05 207 0.034000 0.19490 0.016126 181 5.5750e-05 208 0.033944 0.19487
 0.016124 182 5.5599e-05 209 0.033888 0.19487 0.016124 183 5.1566e-05 210 0.033833 0.19472
 0.016115 184 4.5550e-05 212 0.033730 0.19510 0.016122 185 4.3404e-05 214 0.033639 0.19526
 0.016128 186 4.2536e-05 215 0.033595 0.19530 0.016129 187 3.9508e-05 216 0.033553 0.19613
 0.016187 188 3.7810e-05 217 0.033513 0.19553 0.016148 189 3.7077e-05 218 0.033475 0.19558
 0.016147 190 3.6889e-05 219 0.033438 0.19557 0.016147 191 3.6696e-05 220 0.033401 0.19557
 0.016147 192 3.3334e-05 221 0.033365 0.19558 0.016148 193 3.2199e-05 222 0.033331 0.19567
 0.016149 194 2.7875e-05 223 0.033299 0.19556 0.016140 195 2.7795e-05 224 0.033271 0.19532
 0.016137 196 2.7779e-05 225 0.033243 0.19521 0.016132 197 2.5058e-05 226 0.033216 0.19520
 0.016133 198 2.3457e-05 227 0.033191 0.19558 0.016135 199 2.3342e-05 228 0.033167 0.19544
 0.016134 200 2.2630e-05 229 0.033144 0.19544 0.016134 201 2.1898e-05 230 0.033121 0.19544
 0.016134 202 2.1507e-05 231 0.033099 0.19530 0.016122 203 2.0965e-05 232 0.033078 0.19536
 0.016123 204 2.0395e-05 233 0.033057 0.19532 0.016123 205 1.9845e-05 234 0.033036 0.19526
 0.016123 206 1.9291e-05 235 0.033017 0.19541 0.016127 207 1.8322e-05 237 0.032978 0.19551
 0.016129 208 1.7738e-05 238 0.032960 0.19547 0.016127 209 1.7688e-05 239 0.032942 0.19549
 0.016127 210 1.7662e-05 240 0.032924 0.19549 0.016127 211 1.7469e-05 241 0.032907 0.19549
 0.016127 212 1.7358e-05 242 0.032889 0.19549 0.016127 213 1.7223e-05 243 0.032872 0.19524
 0.016126 214 1.5626e-05 244 0.032855 0.19524 0.016126 215 1.3938e-05 246 0.032823 0.19516
 0.016126 216 1.3206e-05 247 0.032809 0.19535 0.016125 217 1.2756e-05 248 0.032796 0.19535
 0.016125 218 1.1386e-05 249 0.032783 0.19540 0.016125 219 1.0390e-05 250 0.032772 0.19553
 0.016126 220 1.0000e-05 251 0.032762 0.19547 0.016127

CP	nsplit	rel error	xerror	xstd
0.2345	0	1	1.002	0.03161
0.1203	1	0.7655	0.7672	0.02651
0.09169	2	0.6451	0.6914	0.02438
0.04889	3	0.5534	0.5787	0.02053
0.04798	4	0.5046	0.5319	0.02006
0.03991	5	0.4566	0.5026	0.01931
0.0318	6	0.4167	0.4583	0.01822

CP	nsplit	rel error	xerror	xstd
0.03071	7	0.3849	0.4213	0.01761
0.0299	8	0.3542	0.4103	0.01767
0.02628	9	0.3243	0.399	0.01776
0.02139	10	0.298	0.3452	0.0163
0.01492	11	0.2766	0.3148	0.01608
0.01368	13	0.2468	0.2932	0.01533
0.01073	14	0.2331	0.2783	0.01481
0.006732	15	0.2223	0.262	0.01437
0.006728	16	0.2156	0.2555	0.01432
0.006425	17	0.2089	0.2548	0.01432
0.00524	18	0.2025	0.2415	0.01377
0.005156	19	0.1972	0.2372	0.0137
0.005076	21	0.1869	0.2351	0.0137
0.005054	22	0.1818	0.2347	0.01374
0.004946	23	0.1768	0.2323	0.01364
0.004813	24	0.1718	0.231	0.01359
0.004678	25	0.167	0.2277	0.0136
0.004546	26	0.1623	0.2277	0.0136
0.004346	27	0.1578	0.2279	0.01365
0.004002	28	0.1534	0.2228	0.01333
0.003223	29	0.1494	0.2017	0.01241
0.003093	30	0.1462	0.1989	0.01244
0.002992	31	0.1431	0.1954	0.01242
0.002859	32	0.1401	0.1958	0.01253
0.002808	33	0.1373	0.196	0.01263
0.002484	35	0.1317	0.1915	0.01246
0.002394	36	0.1292	0.1914	0.01267
0.002364	38	0.1244	0.1967	0.01403
0.002288	39	0.122	0.1956	0.01402
0.002184	40	0.1197	0.1998	0.01488
0.002081	41	0.1176	0.202	0.01511
0.002029	43	0.1134	0.2037	0.01523
0.002021	44	0.1114	0.2031	0.01522
0.001925	45	0.1093	0.2018	0.01519
0.001699	48	0.1036	0.194	0.01468
0.001646	50	0.1002	0.1949	0.01488
0.001406	52	0.09688	0.1929	0.01481
0.001401	53	0.09547	0.1928	0.01501
0.001352	54	0.09407	0.1926	0.01501
0.001311	55	0.09272	0.1928	0.01504
0.00119	56	0.09141	0.1924	0.01519
0.001179	57	0.09022	0.1937	0.01519
0.00109	58	0.08904	0.1945	0.01527
0.001083	59	0.08795	0.1945	0.01528
0.001081	60	0.08687	0.1946	0.01528
0.001041	61	0.08579	0.1945	0.01529

CP	nsplit	rel error	xerror	xstd
0.001015	62	0.08474	0.1942	0.0153
0.001005	63	0.08373	0.195	0.01538
0.001004	64	0.08272	0.1949	0.01538
0.0009949	65	0.08172	0.1949	0.01538
0.0009453	66	0.08073	0.1945	0.01541
0.0009337	67	0.07978	0.1946	0.01541
0.0008966	68	0.07885	0.196	0.01549
0.000887	69	0.07795	0.197	0.01551
0.0008792	70	0.07706	0.1966	0.0155
0.0008766	71	0.07618	0.1967	0.0155
0.0008687	72	0.07531	0.1965	0.0155
0.0008558	73	0.07444	0.1967	0.01551
0.0007907	74	0.07358	0.1965	0.01555
0.0007614	75	0.07279	0.1961	0.01563
0.0007587	76	0.07203	0.1964	0.01565
0.0007347	77	0.07127	0.1972	0.01568
0.0007151	78	0.07054	0.1962	0.01565
0.0007074	79	0.06982	0.1962	0.01563
7e-04	80	0.06912	0.1948	0.01544
0.0006988	81	0.06842	0.1944	0.01545
0.0006932	82	0.06772	0.1935	0.01544
0.0006817	86	0.06494	0.1932	0.01544
0.0006629	87	0.06426	0.1933	0.01546
0.0006497	88	0.0636	0.1913	0.01533
0.0006119	90	0.0623	0.1913	0.01532
0.000577	92	0.06108	0.1919	0.01536
0.000562	93	0.0605	0.1923	0.01534
0.0005601	94	0.05994	0.1929	0.0154
0.0005555	95	0.05938	0.1929	0.0154
0.0005334	96	0.05882	0.1919	0.01537
0.0005295	97	0.05829	0.1908	0.01537
0.0005289	98	0.05776	0.1908	0.01537
0.0005275	99	0.05723	0.1909	0.01537
0.0005188	100	0.0567	0.1905	0.01537
0.0005109	101	0.05618	0.1903	0.01536
0.0004959	103	0.05516	0.1899	0.01535
0.0004849	104	0.05467	0.1901	0.01546
0.0004796	107	0.05321	0.1914	0.01551
0.0004751	109	0.05225	0.1916	0.01551
0.0004547	110	0.05178	0.1907	0.01551
0.0004515	111	0.05132	0.1901	0.0155
0.0004185	113	0.05042	0.1911	0.01557
0.000417	114	0.05	0.1907	0.01558
0.0004041	115	0.04958	0.1906	0.01558
0.0003849	117	0.04878	0.1897	0.0156
0.0003746	118	0.04839	0.1898	0.01564

CP	nsplit	rel error	xerror	xstd
0.0003137	119	0.04802	0.1885	0.0156
0.0003037	122	0.04708	0.1886	0.01563
0.0003036	123	0.04677	0.1892	0.01564
0.0003011	124	0.04647	0.189	0.01564
0.0002904	125	0.04617	0.1889	0.01565
0.0002859	126	0.04588	0.1888	0.01565
0.0002832	128	0.0453	0.1887	0.01565
0.00028	129	0.04502	0.1887	0.01565
0.0002641	130	0.04474	0.189	0.01566
0.000264	131	0.04448	0.1887	0.01566
0.0002531	132	0.04421	0.1885	0.01566
0.0002489	133	0.04396	0.1887	0.01566
0.0002432	134	0.04371	0.1887	0.01566
0.000222	135	0.04347	0.1892	0.01568
0.000219	136	0.04325	0.1892	0.01569
0.0002179	137	0.04303	0.19	0.01571
0.0002161	138	0.04281	0.1902	0.01571
0.0002089	139	0.04259	0.1926	0.01587
0.0002036	140	0.04238	0.1929	0.01589
0.0002028	141	0.04218	0.1933	0.01589
0.0002017	143	0.04178	0.1932	0.01589
0.0001984	144	0.04157	0.1934	0.01589
0.0001953	145	0.04137	0.1934	0.0159
0.0001927	146	0.04118	0.1934	0.0159
0.0001851	147	0.04099	0.1934	0.0159
0.0001795	149	0.04062	0.1934	0.01592
0.0001735	150	0.04044	0.1932	0.01592
0.0001731	151	0.04026	0.1932	0.01592
0.0001711	152	0.04009	0.1933	0.01593
0.0001656	153	0.03992	0.1933	0.01593
0.0001653	154	0.03975	0.1932	0.01593
0.0001645	155	0.03959	0.1932	0.01593
0.0001628	156	0.03942	0.1932	0.01593
0.0001622	157	0.03926	0.1934	0.01594
0.0001601	158	0.0391	0.1936	0.01595
0.0001593	160	0.03878	0.1935	0.01595
0.0001459	161	0.03862	0.194	0.01597
0.0001436	162	0.03847	0.1949	0.016
0.0001417	163	0.03833	0.1949	0.016
0.0001368	164	0.03819	0.1951	0.016
0.0001337	165	0.03805	0.1949	0.016
0.0001336	166	0.03792	0.1948	0.01597
0.0001309	167	0.03778	0.1949	0.01597
0.0001276	168	0.03765	0.1946	0.01597
0.0001246	170	0.0374	0.1946	0.01597
0.0001241	171	0.03727	0.1946	0.01597

CP	nsplit	rel error	xerror	xstd
0.0001223	172	0.03715	0.1946	0.01597
0.0001198	173	0.03703	0.1942	0.01597
0.0001182	174	0.03691	0.1943	0.01597
0.0001135	175	0.03679	0.1945	0.016
0.000112	176	0.03668	0.1944	0.0161
0.0001111	177	0.03656	0.1946	0.01611
0.0001107	178	0.03645	0.1947	0.01611
0.0001086	179	0.03634	0.1947	0.01611
0.000107	180	0.03623	0.1947	0.01611
0.0001067	181	0.03613	0.1947	0.01611
0.0001051	182	0.03602	0.1948	0.01612
0.0001046	183	0.03591	0.1948	0.01612
0.0001042	185	0.03571	0.1948	0.01612
0.0001036	186	0.0356	0.1948	0.01612
9.98e-05	187	0.0355	0.1947	0.01612
9.971e-05	188	0.0354	0.1946	0.01612
9.881e-05	189	0.0353	0.1946	0.01612
9.375e-05	190	0.0352	0.1948	0.01614
9.261e-05	191	0.03511	0.1952	0.01614
9.058e-05	192	0.03501	0.1952	0.01614
7.834e-05	193	0.03492	0.1955	0.01617
7.713e-05	194	0.03484	0.1956	0.01619
7.507e-05	195	0.03477	0.1951	0.01614
7.168e-05	196	0.03469	0.1951	0.01614
7.133e-05	197	0.03462	0.1949	0.01612
7.045e-05	198	0.03455	0.195	0.01612
6.711e-05	199	0.03448	0.1949	0.01612
6.447e-05	200	0.03441	0.195	0.01612
5.926e-05	201	0.03435	0.1949	0.01613
5.906e-05	202	0.03429	0.1949	0.01612
5.761e-05	203	0.03423	0.195	0.01612
5.707e-05	204	0.03417	0.1949	0.01613
5.704e-05	205	0.03411	0.1949	0.01613
5.702e-05	206	0.03406	0.1949	0.01613
5.575e-05	207	0.034	0.1949	0.01613
5.575e-05	208	0.03394	0.1949	0.01612
5.56e-05	209	0.03389	0.1949	0.01612
5.157e-05	210	0.03383	0.1947	0.01611
4.555e-05	212	0.03373	0.1951	0.01612
4.34e-05	214	0.03364	0.1953	0.01613
4.254e-05	215	0.0336	0.1953	0.01613
3.951e-05	216	0.03355	0.1961	0.01619
3.781e-05	217	0.03351	0.1955	0.01615
3.708e-05	218	0.03348	0.1956	0.01615
3.689e-05	219	0.03344	0.1956	0.01615
3.67e-05	220	0.0334	0.1956	0.01615

CP	nsplit	rel error	xerror	xstd
3.333e-05	221	0.03336	0.1956	0.01615
3.22e-05	222	0.03333	0.1957	0.01615
2.788e-05	223	0.0333	0.1956	0.01614
2.78e-05	224	0.03327	0.1953	0.01614
2.778e-05	225	0.03324	0.1952	0.01613
2.506e-05	226	0.03322	0.1952	0.01613
2.346e-05	227	0.03319	0.1956	0.01614
2.334e-05	228	0.03317	0.1954	0.01613
2.263e-05	229	0.03314	0.1954	0.01613
2.19e-05	230	0.03312	0.1954	0.01613
2.151e-05	231	0.0331	0.1953	0.01612
2.097e-05	232	0.03308	0.1954	0.01612
2.04e-05	233	0.03306	0.1953	0.01612
1.985e-05	234	0.03304	0.1953	0.01612
1.929e-05	235	0.03302	0.1954	0.01613
1.832e-05	237	0.03298	0.1955	0.01613
1.774e-05	238	0.03296	0.1955	0.01613
1.769e-05	239	0.03294	0.1955	0.01613
1.766e-05	240	0.03292	0.1955	0.01613
1.747e-05	241	0.03291	0.1955	0.01613
1.736e-05	242	0.03289	0.1955	0.01613
1.722e-05	243	0.03287	0.1952	0.01613
1.563e-05	244	0.03285	0.1952	0.01613
1.394e-05	246	0.03282	0.1952	0.01613
1.321e-05	247	0.03281	0.1953	0.01612
1.276e-05	248	0.0328	0.1953	0.01612
1.139e-05	249	0.03278	0.1954	0.01612
1.039e-05	250	0.03277	0.1955	0.01613
1e-05	251	0.03276	0.1955	0.01613

We can see from the results above that, in this case, the CV error starts to rise when $CP = 0.0046777$.

BUT, there is a standard error in that point estimate! If we do $0.22768 + 0.013597 = 0.241277$

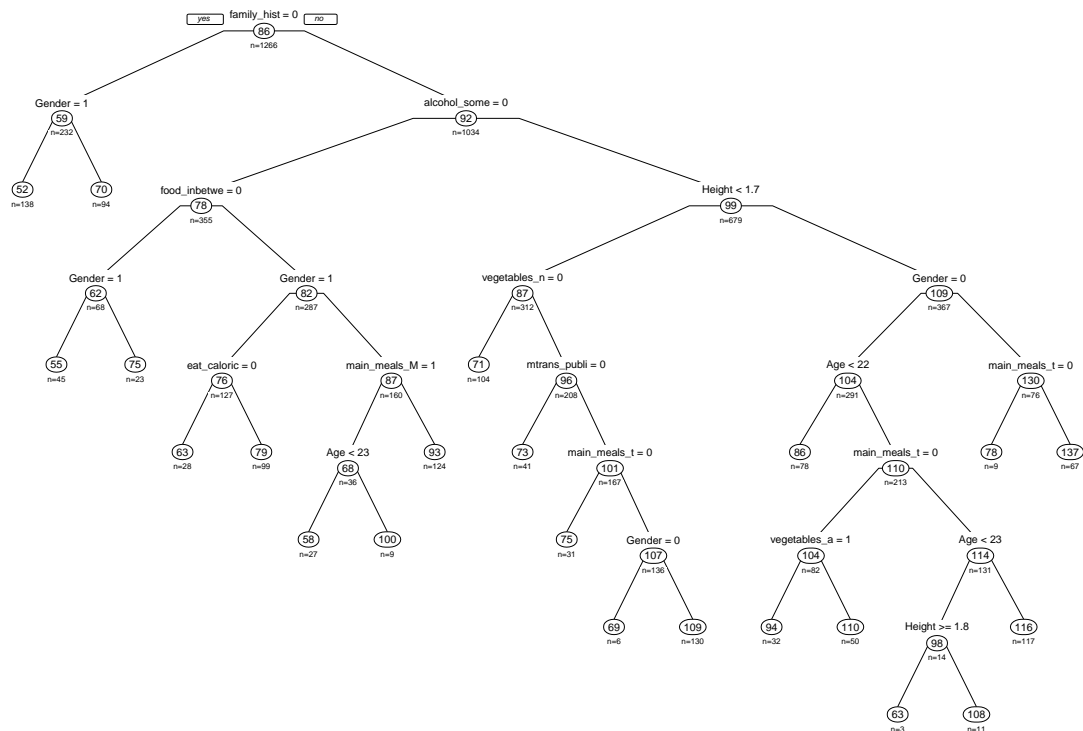
So, we can go for a SMALLER (and thus better) tree with 19 splits instead of 24, which corresponds to a CP of 0.0051561.

So now we will fit the FINAL prediction tree with a CP of 0.0051561, which is the best value for CP because we calculated it with a Cross-Validation approach!

```
set.seed(1)

tree_2 <- rpart(Weight ~ ., data = train.set,
  method = "anova", control = rpart.control(cp = 0.0051561,
    minbucket = 1, maxdepth = 10))
```

```
plot_tree = prp(tree_2, type = 1, extra = 1, under = TRUE,
  split.font = 1, varlen = -10)
```



Now, let's compare the RMSE for validation and training sets.

```
# First, let's create two vectors, one for the
# predicted values, and another for the actual
# values :
```

```
predicted_train <- predict(tree_2, train.set)
```

```
actual_train <- train.set$Weight
```

```
# And lastly, we make use of the RSME formula
# to calculate it :
```

```
RMSE_train = sqrt(mean((predicted_train - actual_train)^2))
```

```
RMSE_train
```

```
## [1] 11.29379
```

We have RMSE = 11.29379

Now, we do the same but for the validation set.

```
predicted_valid <- predict(tree_2, valid.set)

actual_valid <- valid.set$Weight

RMSE_valid = sqrt(mean((predicted_valid - actual_valid)^2))

RMSE_valid

## [1] 13.25937
```

The RMSE for the validation data is 13.25937.

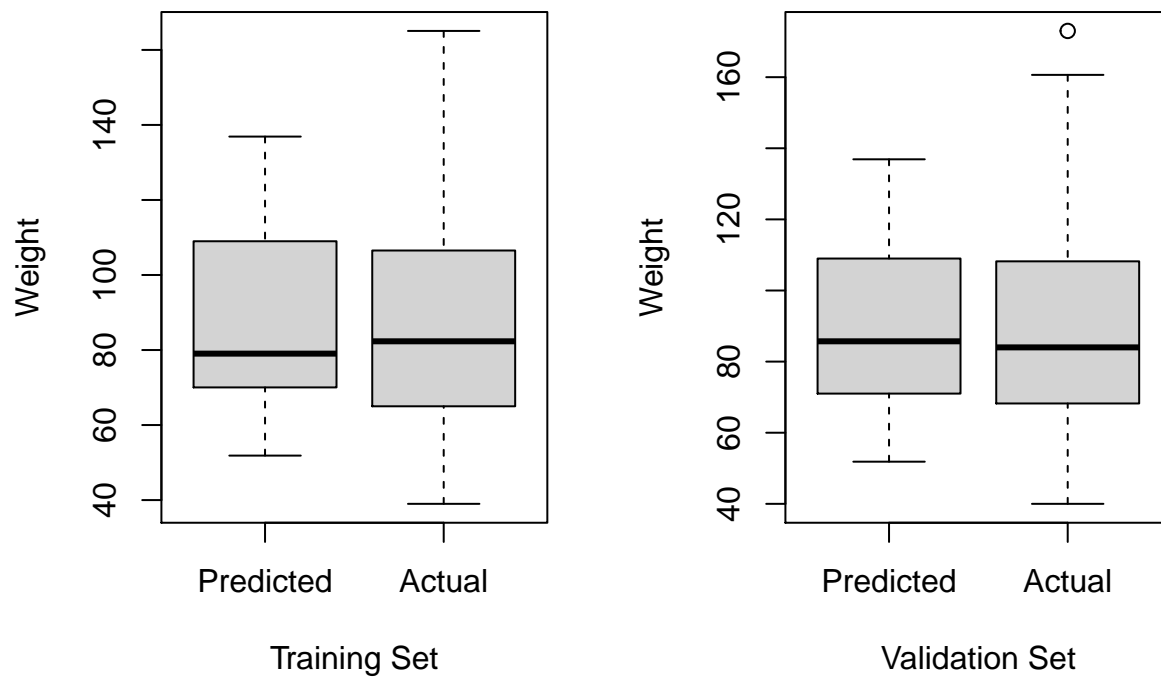
It is very normal that RMSE is smaller with the training data, because we have selected the optimal CP according to the training data. However, the difference seems not so big.

Let's now look at some boxplots to compare the performance of the tree on both sets (training and validation).

```
par(mfrow = c(1, 2))

boxplot(predicted_train, actual_train, names = c("Predicted",
  "Actual"), ylab = "Weight", xlab = "Training Set")

boxplot(predicted_valid, actual_valid, names = c("Predicted",
  "Actual"), ylab = "Weight", xlab = "Validation Set")
```



It is difficult to judge on which set the tree has performed better. Probably the higher RMSE for the validation set is due to the presence of an outlier!

But the training set boxplot seems a bit right skewed, so one could conclude that the validation set did even a better job.

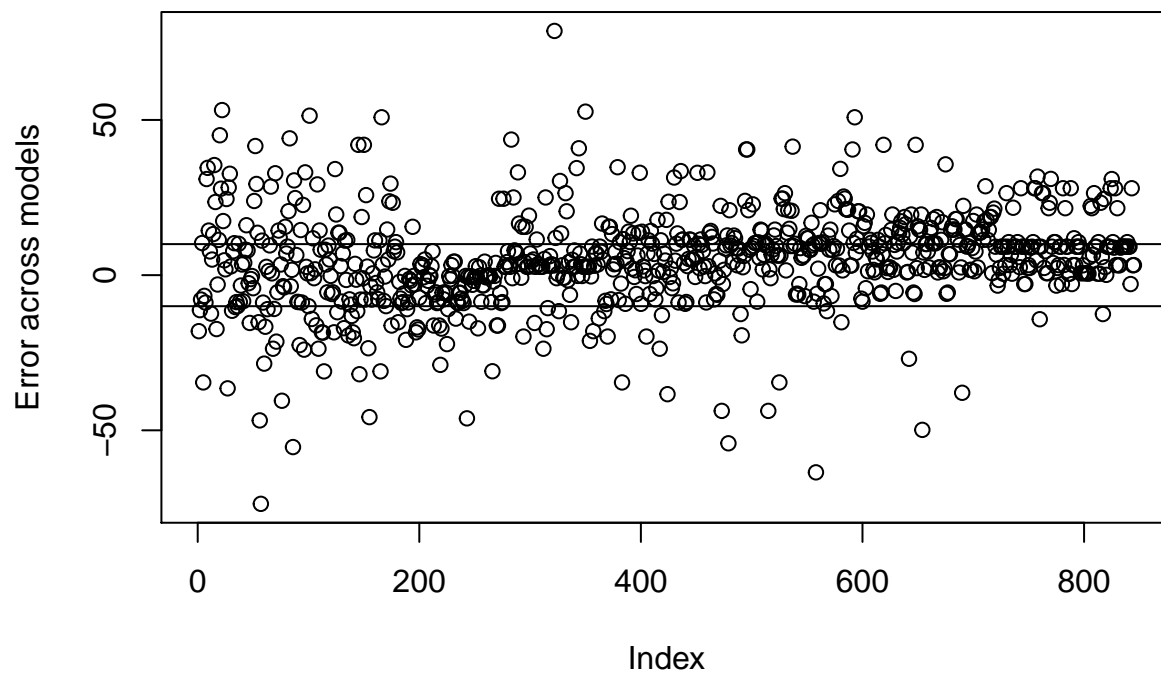
So now we are finished with the regression tree.

We can leave the CP as it is, and the tree will grow until 19 splits.

Now, let's do something quite interesting! We will do a comparison of both KNN and regression tree on the validation set.

We will plot the errors "across the models", so the difference between the predicted weights by both models.

```
plot(predicted - predicted_valid, ylab = "Error across models")
abline(h = 10)
abline(h = -10)
```



We can see that although there is quite a lot of variance, at times both models seem to behave almost equally at predicting the weight.

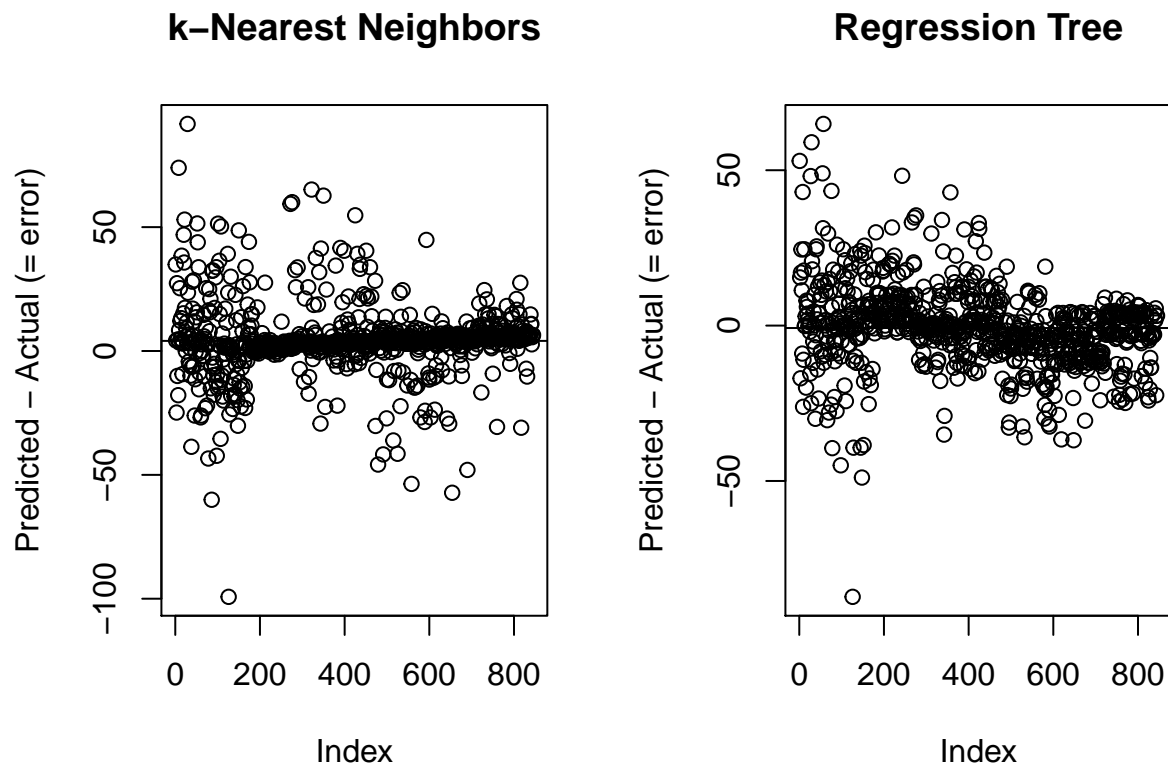
Inside the range of $[-10 ; 10]$ there seems to be the majority of the points, so the range is not so big.

Let's take a more precise look at each method compared with the validation data.

```
par(mfrow = c(1, 2))

# For KNN :
plot(predicted - valid.set[, 4], main = "k-Nearest Neighbors",
     ylab = "Predicted - Actual (= error)")
abline(h = mean(predicted - valid.set[, 4]))

# For tree :
plot(predicted_valid - valid.set[, 4], main = "Regression Tree",
     ylab = "Predicted - Actual (= error)")
abline(h = mean(predicted_valid - valid.set[,
     4]))
```



VERY INTERESTING!

Actually we can see that we have the typical **trade-off between BIAS and VARIANCE**.

Certainly, the k-NN seems to be more precise (less variance), since the points are less far apart from each other. BUT, we observe an upward trend, and the mean of the points (= errors) is at around 4, not 0!

So there is small bias in the k-NN.

However, the regression tree has a lot of variance, BUT on average it is very precise (the mean of the errors is almost at zero). Indeed :

```
# For k-NN :
```

```
mean(predicted - valid.set[, 4])
```

```
## [1] 4.079884
```

```
# For tree :
```

```
mean(predicted_valid - valid.set[, 4])
```

```
## [1] -0.7759031
```

For the regression tree, the mean is very close to zero.

So now we can discuss which model is better for us. Do we want a very accurate prediction although it may be around on average 4 Kg away from the truth? Or do we want a prediction which is very far from the truth but, taking into account all predictions, on average we are almost exactly on the target?

Probably we want something like the k-NN, since 4 Kg is not much.

The regression tree is less biased and has smaller RMSE, BUT certainly the amount of error is very big (the differences predicted - actual are quite big).

Therefore, we may prefer the k-NN after all!

Ensemble Method (MLR + k-NN + Regression Tree)

The aim of this ensemble method is to combine the Multiple Linear Regression, the k-NN and the Regression Tree in order to obtain even better results. This combination of methods will be done by taking the average prediction of the variable of interest (**Weight**).

This means that the predicted weight using this ensemble method will be obtained by running the three methods separately and then taking the average over the results.

Creating dataframe :

```
ensemble_df <- data.frame(actual = valid.set[,  
  4], MLR = backward_pred_obesity, knn = predicted,  
  Regression_tree = predicted_valid, Ensemble_Method = (predicted +  
    predicted_valid + backward_pred_obesity)/3)  
  
head(ensemble_df)
```

##	actual	MLR	knn	Regression_tree	Ensemble_Method
## 2	56	63.92756	90.93895	108.99283	87.95311
## 3	77	94.05396	81.22554	92.50013	89.25988
## 4	87	82.71985	62.15660	70.03409	71.63685
## 6	53	59.01734	80.37579	70.03409	69.80907
## 10	68	87.56165	57.93629	92.50013	79.33269
## 11	105	101.38382	109.66225	116.34191	109.12933

Creating a person for prediction


```

# Creating a person for prediction :

example.df = obesity[1, ]

example.df$Gender = "Male"
example.df$Age = 25
example.df$Height = 1.78
example.df$Weight = 70
example.df$family_history = "no"
example.df$eat_caloric = "no"
example.df$vegetables = "Always"
example.df$main_meals = "More_than_3"
example.df$food_inbetween = "no"
example.df$SMOKE = "no"
example.df$CH2O = "Between 1 and 2 L"
example.df$monitor_cal = "no"
example.df$physical_act = "2 or 4 days"
example.df$tech_devices = "0-2_hours"
example.df$alcohol = "no"
example.df$MTRANS = "Walking"
example.df$NObeyesdad = "Normal_weight"

norm.values <- preProcess(obesity[, c(2:4)], method = "range")

example.norm <- predict(norm.values, example.df)

example.df = to_factor(example.df)

# example.df = dummy(example.df) Create a
# SINGLE function for dummyfication (need to
# eliminate the _dummy dataset)

# predict(k_nn, example.df) predict(tree_2,
# example.df) ...

# for Multiple Linear Regression, simply
# replace coefficients and variable values
# into the formula to get the predicted
# weight. Maybe create a function called
# 'MLR()' for this!

```