Data Mining Project (MaBAn 2020)

Predicting obesity levels according to daily habits

by : Ángel Tomás-Ripoll & Laurence Tétreault-Falsafi

Contents

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 :

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

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

```r
# Used packages :
library(pander)
library(dplyr)
library(gt)
library(car)
library(ggplot2)
library(gridExtra)
library(psych)
```

```r
library(corrplot)
library(ellipse)
library(dummies)
library(nnet)

# Working Directory :
setwd("~/GitHub/CVTDM_Project_MaBAn_2020")


# Reading the data :
obesity <- read.csv("Obesity.csv", header = T,
    sep = ",")
attach(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 :

```r
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 the fourth one, the "Weight", so it will be our dependent variable.

We were "lucky" on the fact that this dataset has a quite high level of quality, because it has no missing observations, and our subsequent exploratory analysis will tell us if there are outliers to be handled with.

Once we are done with a Data Exploratory Analysis and with a proper Data Pre-Processing, 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** (not ANOVA since "Age" and "Height" are numerical)
2. **Classification tree** (complemented with a random forest / boosted trees / bagged trees)
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 comming 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!

```
# 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"
```

```r
# 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" "Overweight_Level_I" …

```r
pander(summary(obesity))
```

Table continues below

Gender	Age	Height	Weight
Length:2111	Min. :14.00	Min. :1.450	Min. : 39.00
Class :character	1st Qu.:19.95	1st Qu.:1.630	1st Qu.: 65.47
Mode :character	Median :22.78	Median :1.700	Median : 83.00
NA	Mean :24.31	Mean :1.702	Mean : 86.59
NA	3rd Qu.:26.00	3rd Qu.:1.768	3rd Qu.:107.43
NA	Max. :61.00	Max. :1.980	Max. :173.00

Table continues below

family_history	eat_caloric	vegetables	main_meals
Length:2111	Length:2111	Min. :1.000	Min. :1.000
Class :character	Class :character	1st Qu.:2.000	1st Qu.:2.659
Mode :character	Mode :character	Median :2.386	Median :3.000
NA	NA	Mean :2.419	Mean :2.686
NA	NA	3rd Qu.:3.000	3rd Qu.:3.000
NA	NA	Max. :3.000	Max. :4.000

Table continues below

food_inbetween	SMOKE	CH2O	monitor_cal
Length:2111	Length:2111	Min. :1.000	Length:2111
Class :character	Class :character	1st Qu.:1.585	Class :character
Mode :character	Mode :character	Median :2.000	Mode :character
NA	NA	Mean :2.008	NA
NA	NA	3rd Qu.:2.477	NA
NA	NA	Max. :3.000	NA

Table continues below

physical_act	tech_devices	alcohol	MTRANS
Min. :0.0000	Min. :0.0000	Length:2111	Length:2111
1st Qu.:0.1245	1st Qu.:0.0000	Class :character	Class :character

physical_act	tech_devices	alcohol	MTRANS
Median :1.0000	Median :0.6253	Mode :character	Mode :character
Mean :1.0103	Mean :0.6579	NA	NA
3rd Qu.:1.6667	3rd Qu.:1.0000	NA	NA
Max. :3.0000	Max. :2.0000	NA	NA

NObeyesdad
Length:2111
Class :character
Mode :character
NA
NA
NA

Now, since many variables are in fact 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 refered to earlier (https://www.sciencedirect.com/science/article/pii/S2352340919306985).

```
# Binning some numerical variables :


# vegetables

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

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

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


# main_meals

obesity$main_meals[obesity$main_meals >= 1 & obesity$main_meals <
    3] <- "Btw_1_&_2"

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

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


# tech_devices

obesity$tech_devices[obesity$tech_devices >= 0 &
    obesity$tech_devices <= 2] <- "0-2_hours"
```

```r
obesity$tech_devices[obesity$tech_devices >= 3 &
    obesity$tech_devices <= 5] <- "3-5_hours"

obesity$main_meals[obesity$tech_devices > 5] <- "more_than_5_hours"


# physical_act

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

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

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

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


# CH2O

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

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

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

As we saw with the `str()` function, all the categorical variables are treated as `character`.

Therefore, we will now convert all the categorical variables to `factor` type.

```r
# Converting character variables to factor :

obesity$Gender = as.factor(obesity$Gender)
obesity$family_history = as.factor(obesity$family_history)
obesity$eat_caloric = as.factor(obesity$eat_caloric)
obesity$food_inbetween = as.factor(obesity$food_inbetween)
obesity$SMOKE = as.factor(obesity$SMOKE)
obesity$monitor_cal = as.factor(obesity$monitor_cal)
obesity$alcohol = as.factor(obesity$alcohol)
obesity$MTRANS = as.factor(obesity$MTRANS)
obesity$NObeyesdad = as.factor(obesity$NObeyesdad)
obesity$vegetables = as.factor(obesity$vegetables)
obesity$main_meals = as.factor(obesity$main_meals)
obesity$CH2O = as.factor(obesity$CH2O)
obesity$physical_act = as.factor(obesity$physical_act)
obesity$tech_devices = as.factor(obesity$tech_devices)
```

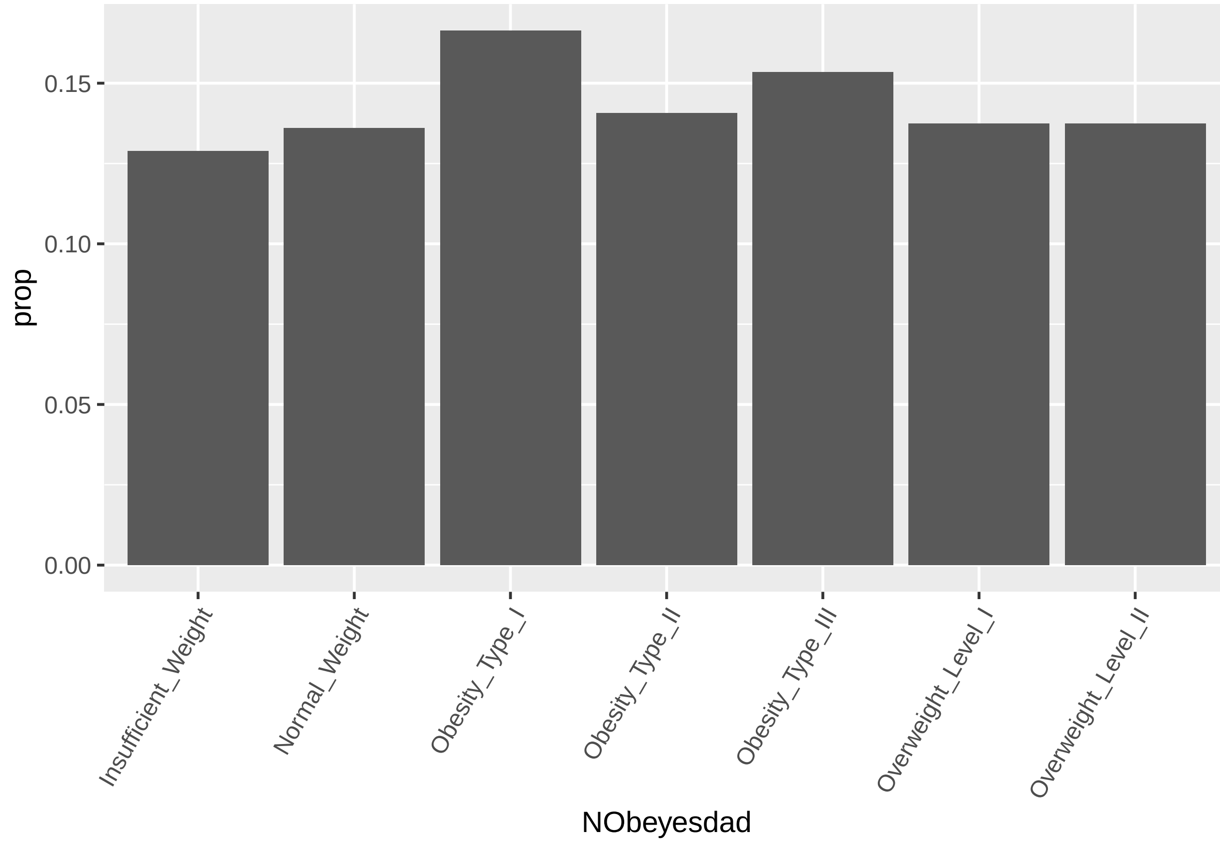
Let's now remove any missing values.

```
# Removing Missing values

na.omit(obesity)
```

There were no missing values within our dataset.

Exploratory Data Analysis

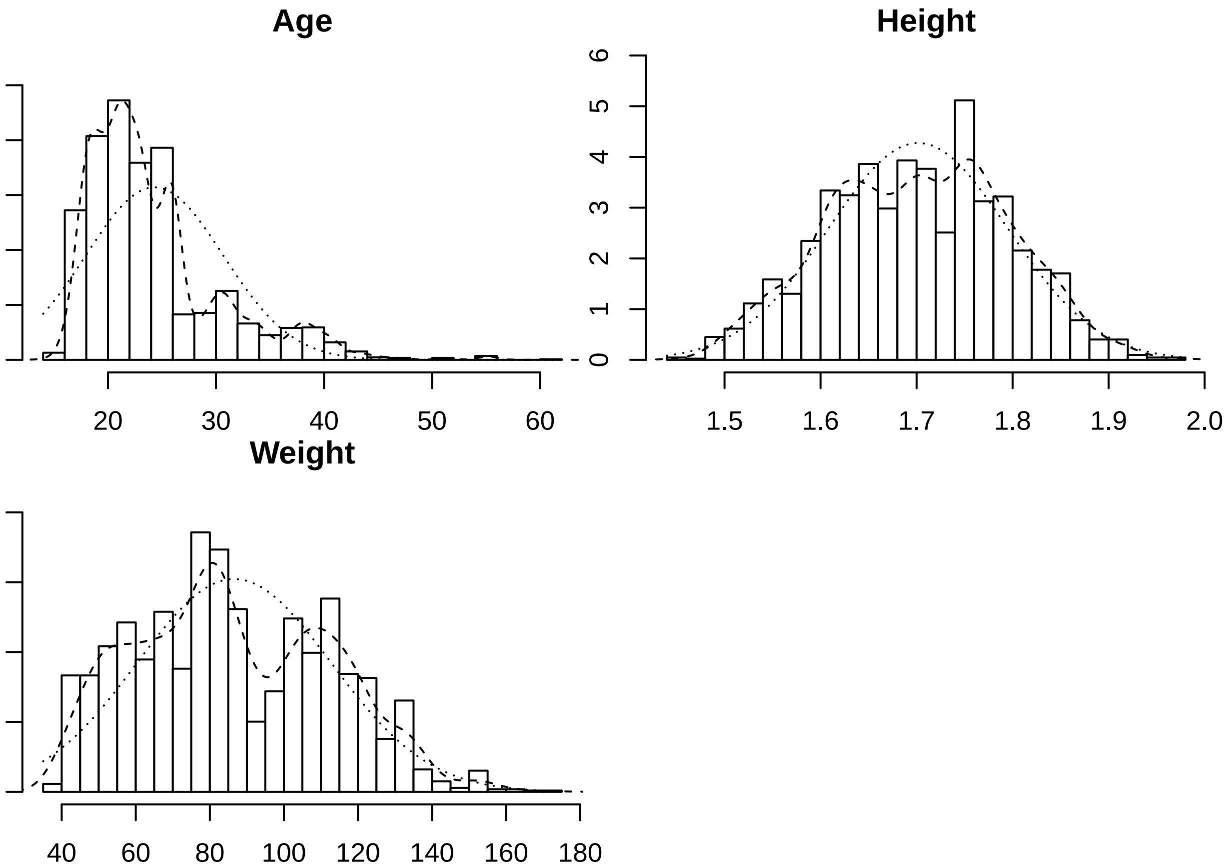
```
ggplot(data = obesity, aes(x = NObeyesdad)) +
    geom_bar(aes(y = ..prop.., group = 1)) + theme(axis.text.x = element_text(angle = 60,
    hjust = 1))
```



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.

```r
# Creating histograms :

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



#Interpretation:

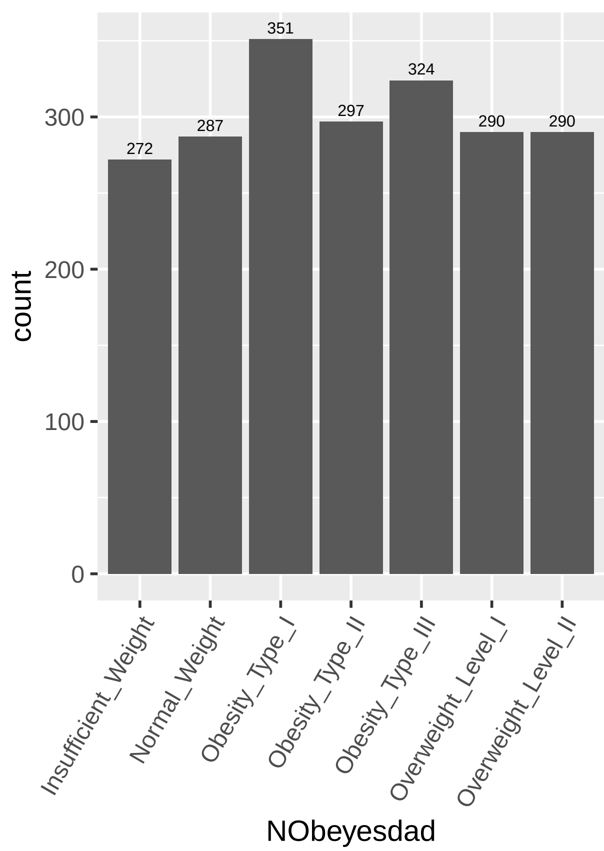
...

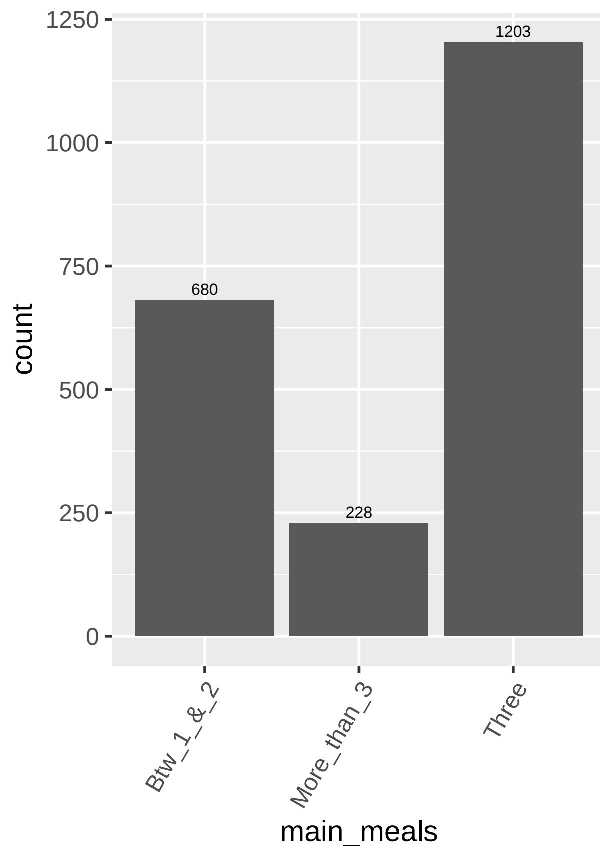
Now, let's do some barplots in order to get an idea of the distribution of each of the categorical variables.

```
# 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)
```

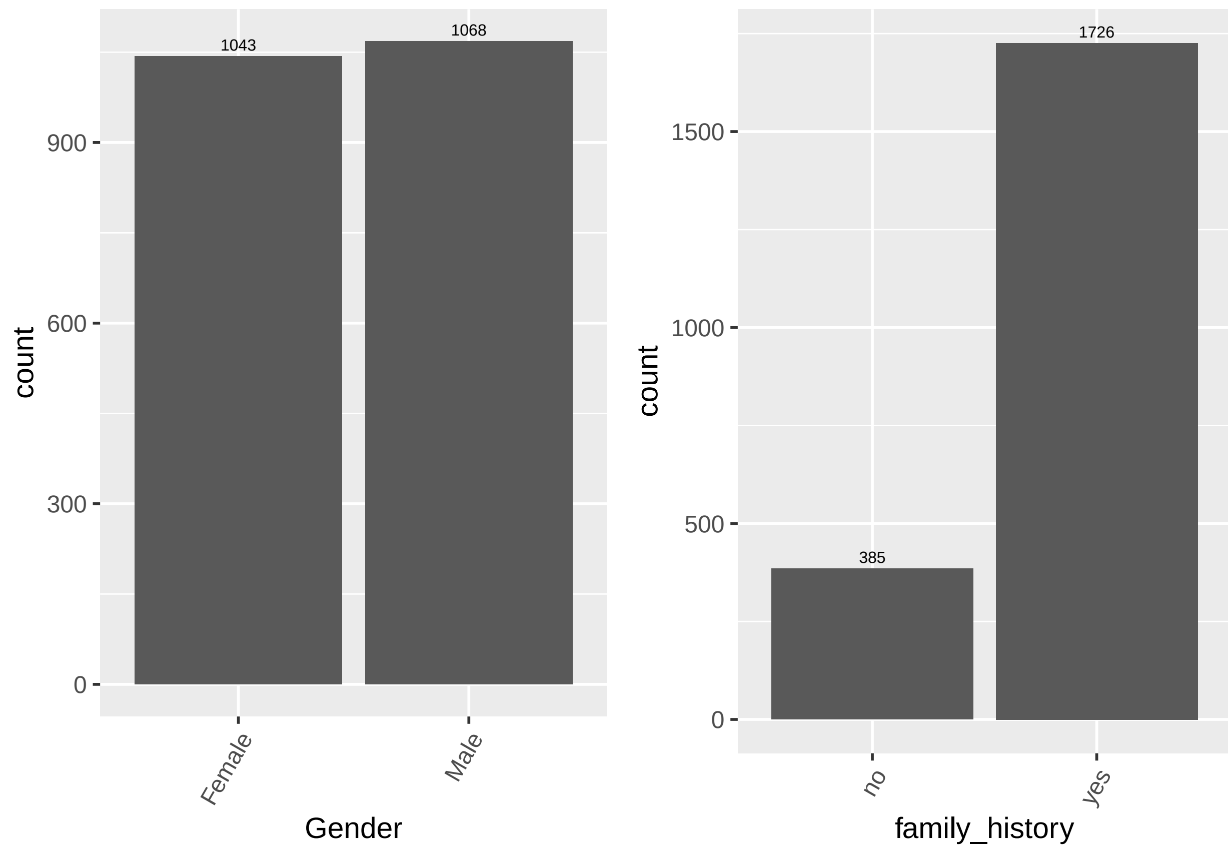
```r
plot_4 = ggplot(data = obesity, aes(x = family_history)) +
    geom_bar(aes(y = ..count.., group = 1)) +
    theme(axis.text.x = element_text(angle = 60,
        hjust = 1)) + geom_text(stat = "count",
    aes(label = ..count..), vjust = -0.5, size = 2.2)

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 = CH2O)) +
    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)
```

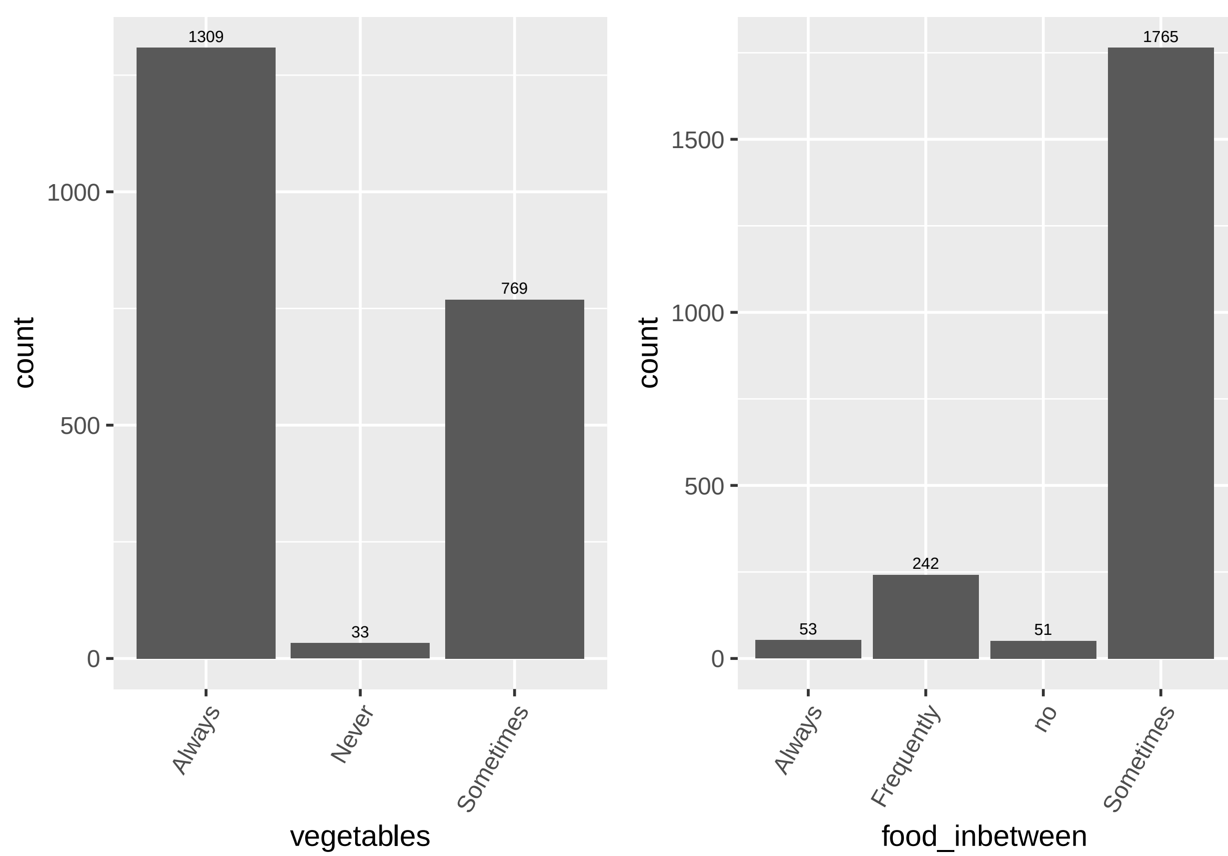




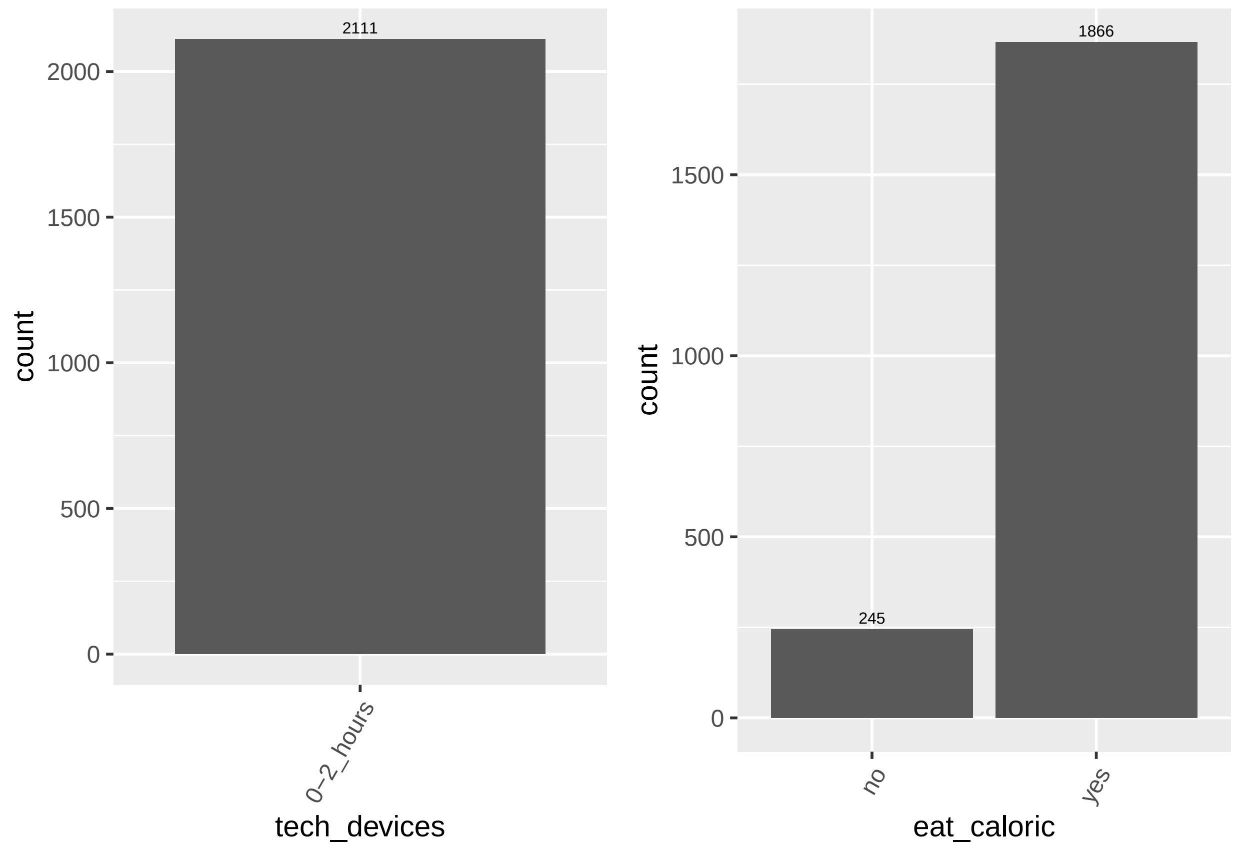

```
grid.arrange(plot_3, plot_4, ncol = 2)
```



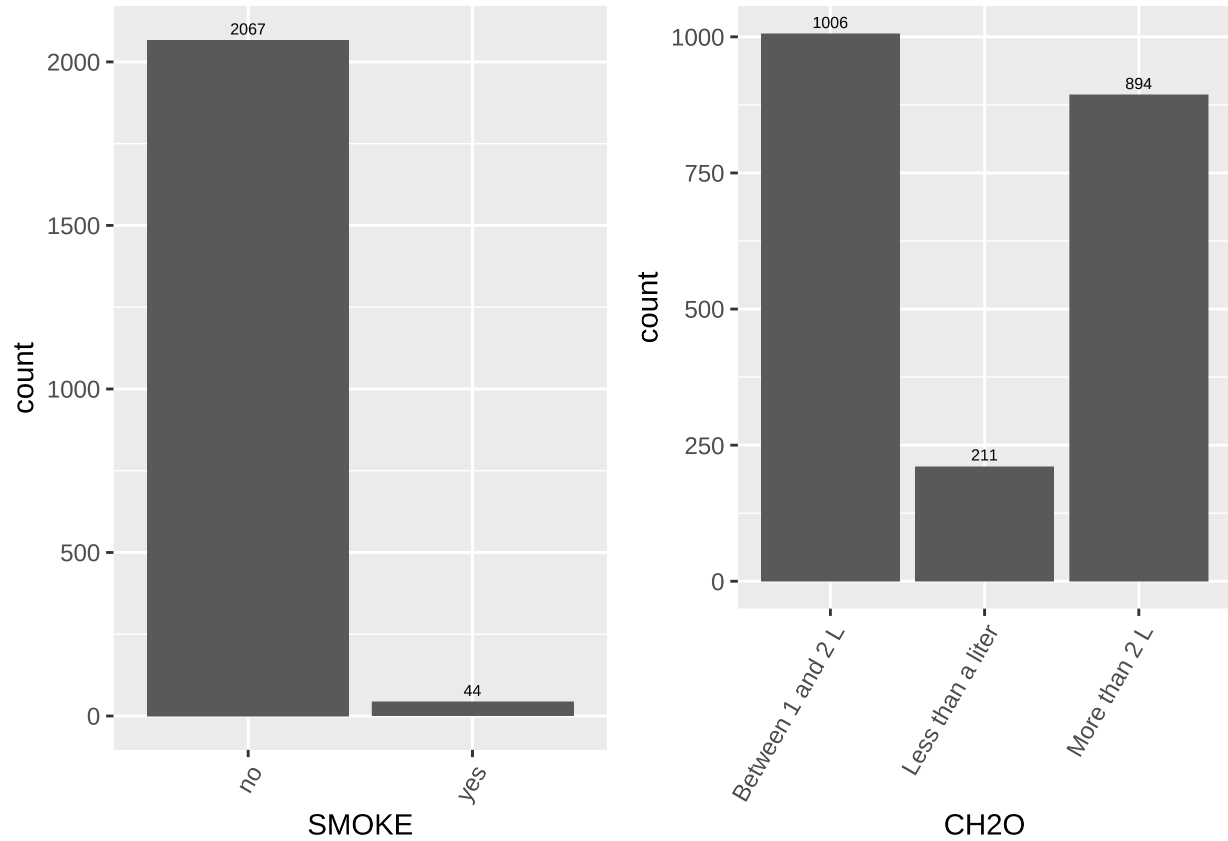

```
grid.arrange(plot_5, plot_6, ncol = 2)
```



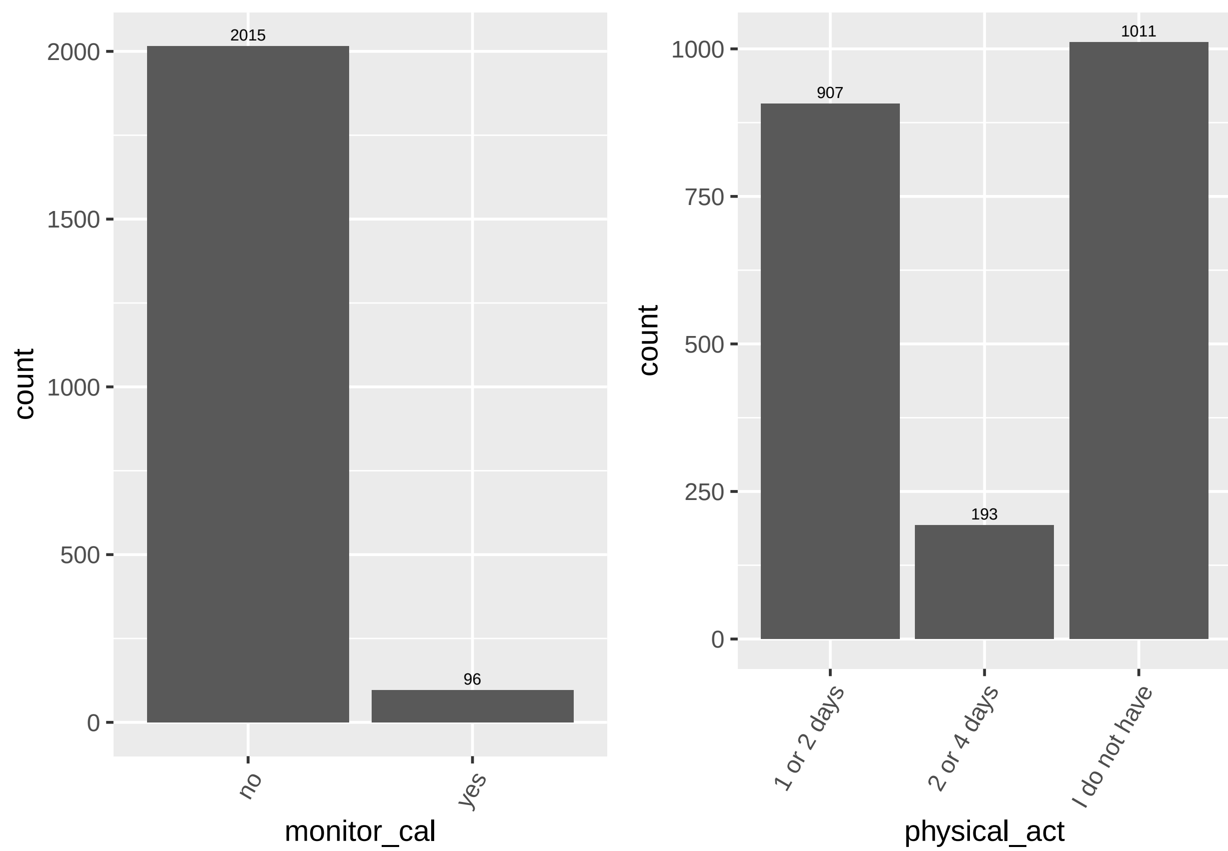
```r
grid.arrange(plot_7, plot_8, ncol = 2)
```



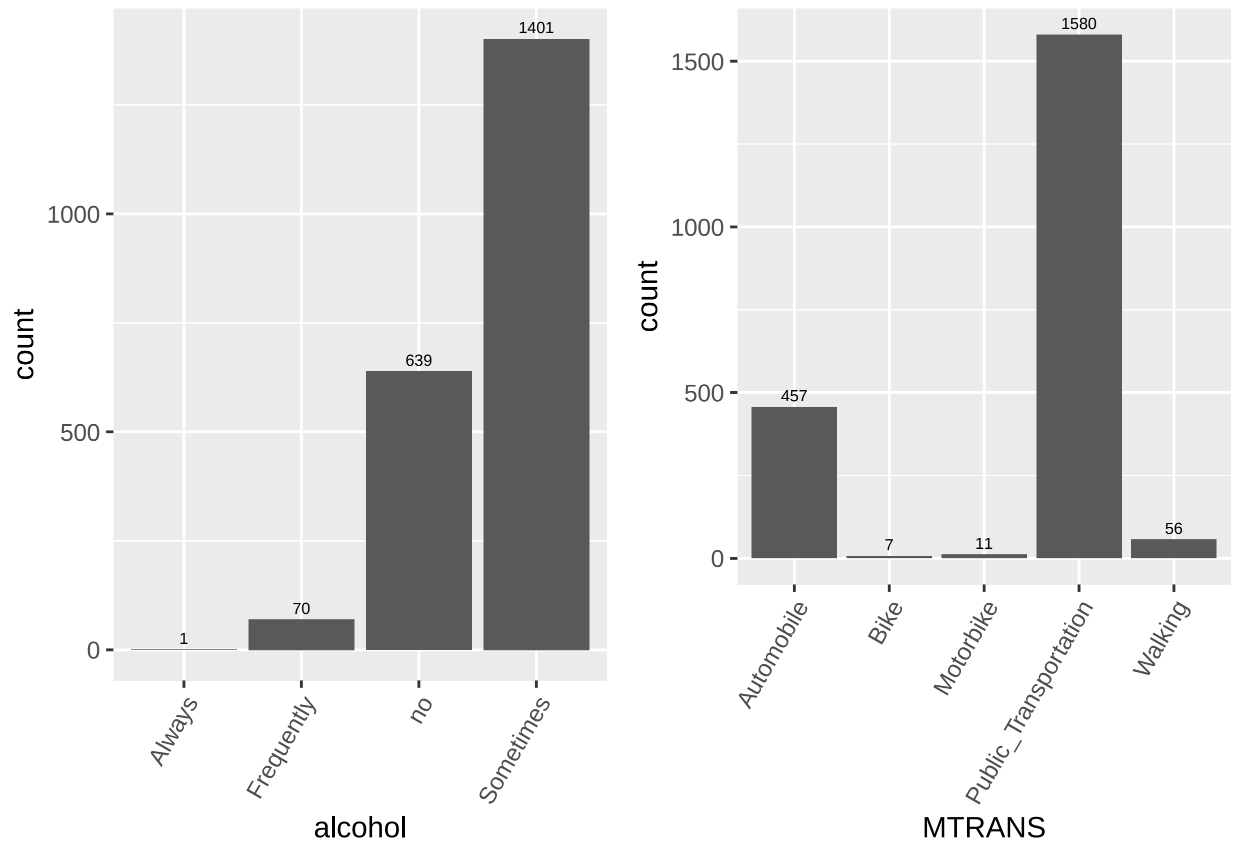

```r
grid.arrange(plot_9, plot_10, ncol = 2)
```




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

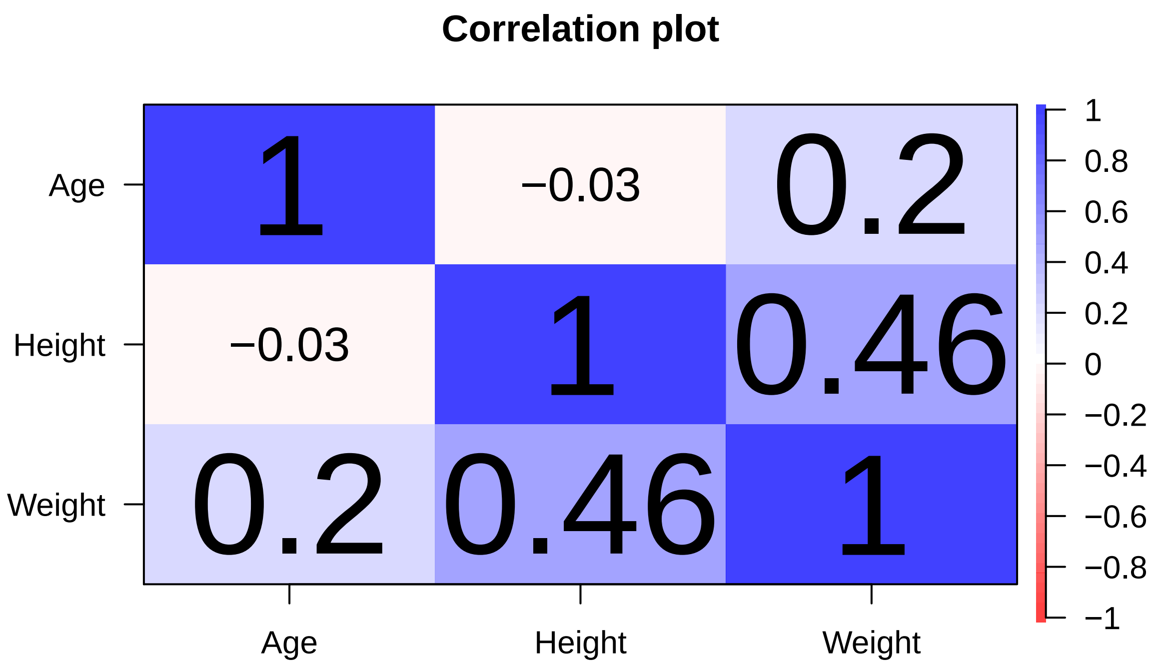


```r
grid.arrange(plot_13, plot_14, ncol = 2)
```



Let's look at the correlations between the variables.

```r
# Correlation plot

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



And now we dummify all the categorical and binary variables, in order to make them "ready" for the subsequent data analysis!

```
# 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
```

```r
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
```

```r
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
```

```r
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
```

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



# Dummmyfying 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
```

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

```r
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
```

```r
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
```

```r
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
```

```r
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
```

```r
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
```

```r
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 a 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])
names(obesity_dum)[25] <- c("tech_devices_0_2")

# NObeyesdad
obesity_dum <- cbind(obesity_dum[1:34], dummy(obesity_dum$NObeyesdad,
    sep = "_"))
```

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

```r
names(obesity_dum)[35:41] <- c("insufficient_weight",
    "normal_weight", "obesity_type_1", "obesity_type_2",
    "obesity_type_3", "overweight_level_1", "overweight_level_2")
```

Data Analysis

Multiple Linear Regression

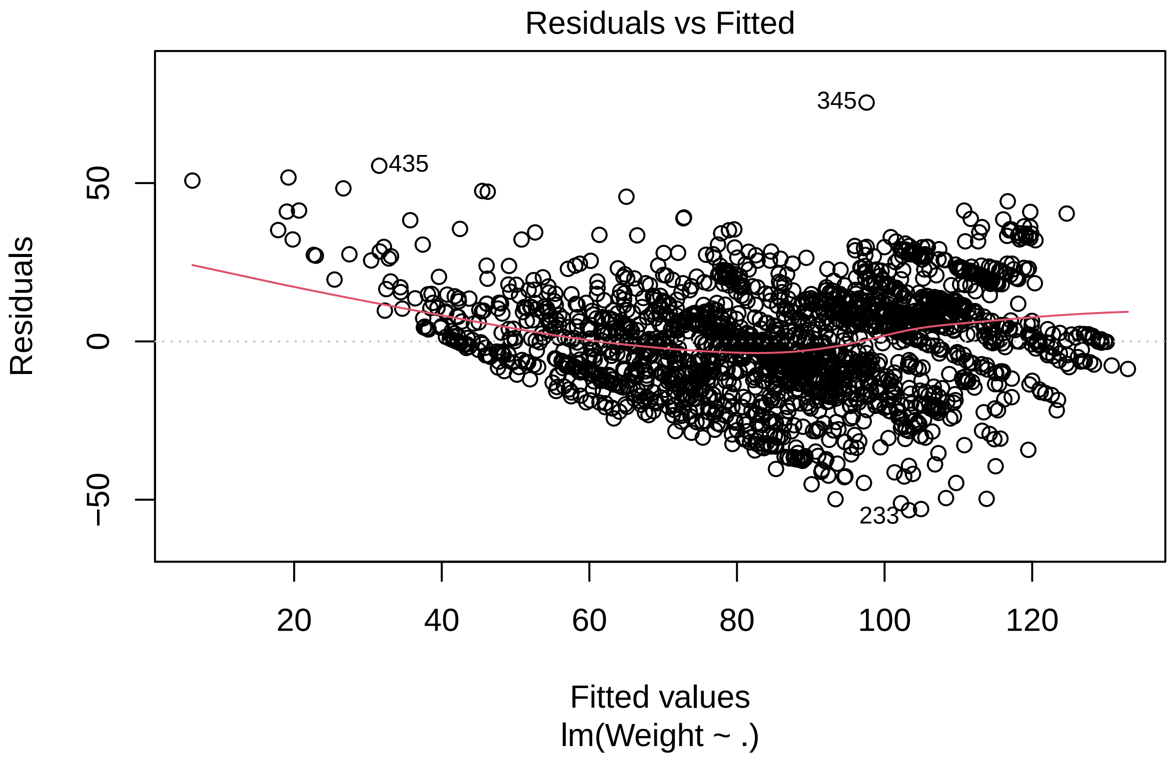
```r
obesity_lm_dum <- subset(obesity_dum, select = c(1:34))

# Linear regression

lm_weight <- lm(Weight ~ ., data = obesity_lm_dum)
summary(lm_weight)
```

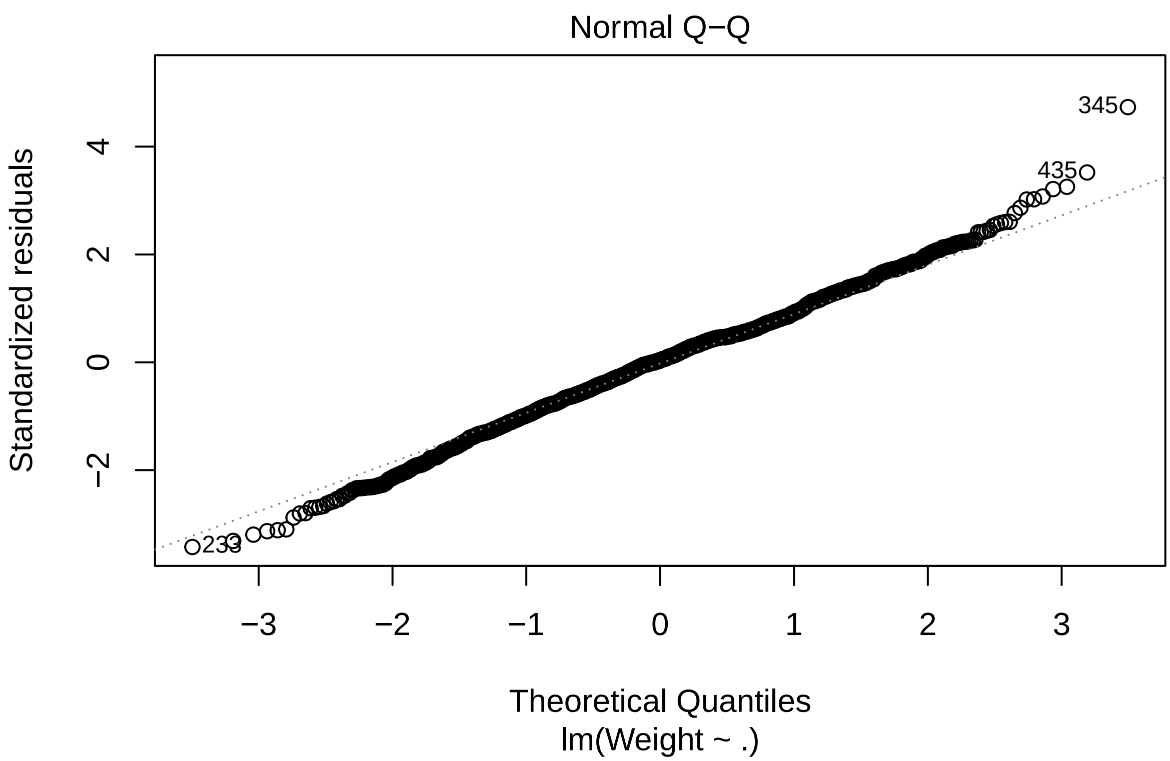
```
##
## Call:
## lm(formula = Weight ~ ., data = obesity_lm_dum)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.312 -10.146   0.605   9.470  75.435
##
## Coefficients: (8 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -149.08908   10.29480 -14.482  < 2e-16 ***
## Gender                        4.37505    0.97478   4.488 7.57e-06 ***
## Age                           0.80340    0.07495  10.719  < 2e-16 ***
## Height                      116.63426    5.58341  20.889  < 2e-16 ***
## family_hist                  16.00171    1.05450  15.175  < 2e-16 ***
## eat_caloric                   4.34834    1.18617   3.666 0.000253 ***
```

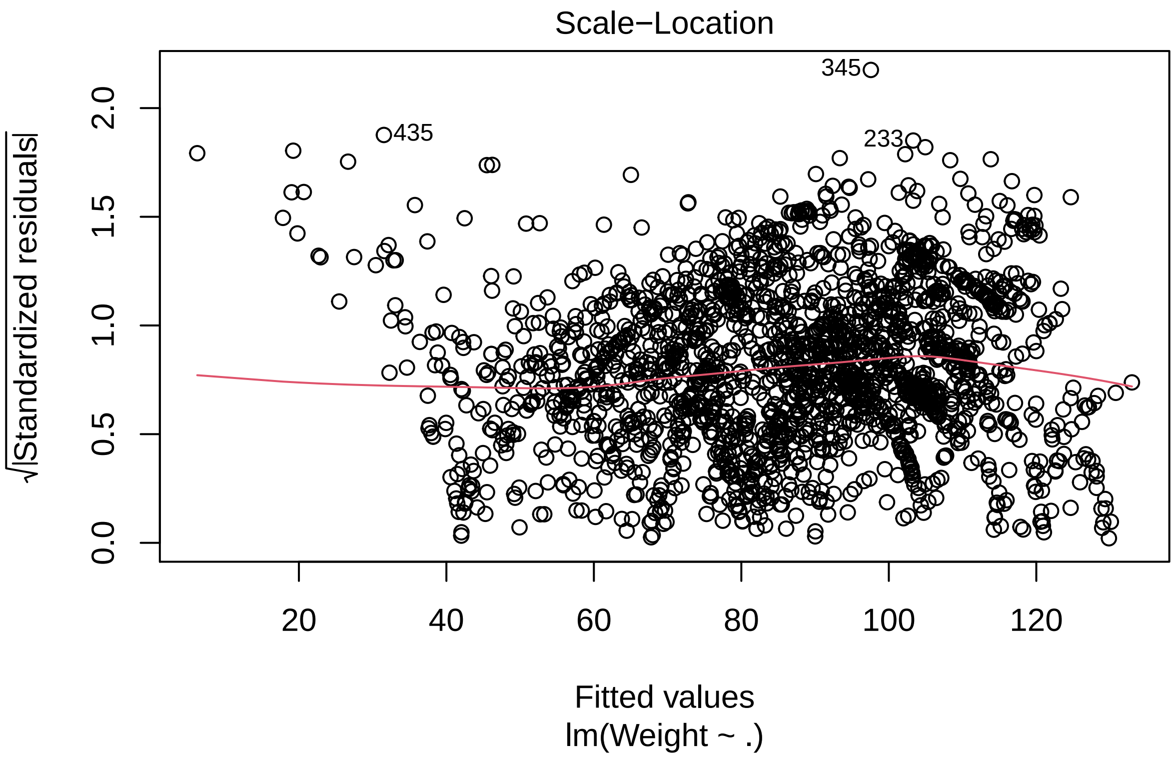
```
## vegetables_never               7.72270    0.78266   9.867  < 2e-16 ***
## vegetables_sometimes          -1.41353    2.90861  -0.486 0.627032
## vegetable_always                   NA         NA      NA       NA
## `main_meals_Btw_1_&_2`        -5.50913    0.81450  -6.764 1.74e-11 ***
## main_meals_More_than_3       -19.23989    1.22675 -15.684  < 2e-16 ***
## main_meals_three                   NA         NA      NA       NA
## food_inbetween_always         -6.80923    2.30144  -2.959 0.003124 **
## food_inbetween_frequently    -16.66256    1.21416 -13.724  < 2e-16 ***
## food_inbetween_no             -1.38766    2.43871  -0.569 0.569408
## food_inbetween_sometimes           NA         NA      NA       NA
## smoke                          0.19506    2.49692   0.078 0.937738
## CH2O_less_than_a_liter        -5.79054    0.78148  -7.410 1.83e-13 ***
## CH2O_between_1_and_2          -5.50868    1.35109  -4.077 4.73e-05 ***
## CH2O_more_than_2                   NA         NA      NA       NA
## monitor_cal                   -6.15091    1.76383  -3.487 0.000498 ***
## physical_act_do_not_have      -0.25281    0.78944  -0.320 0.748815
## physical_act_1_2              -9.52005    1.33865  -7.112 1.57e-12 ***
## physical_act_2_4                   NA         NA      NA       NA
## tech_devices_0_2                   NA         NA      NA       NA
## alcohol_always                 6.80451   16.26022   0.418 0.675642
## alcohol_frequently            -5.24417    2.02287  -2.592 0.009596 **
## alcohol_no                    -4.63135    0.83277  -5.561 3.02e-08 ***
## alcohol_sometimes                  NA         NA      NA       NA
## mtrans_automobile             -2.98080    2.44138  -1.221 0.222244
## mtrans_bike                   -1.13247    6.47042  -0.175 0.861079
## mtrans_motorbike               7.75960    5.34784   1.451 0.146936
## mtrans_public_transportation   8.71996    2.26794   3.845 0.000124 ***
## mtrans_walking                     NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.02 on 2085 degrees of freedom
## Multiple R-squared:  0.6305, Adjusted R-squared:  0.626
## F-statistic: 142.3 on 25 and 2085 DF,  p-value: < 2.2e-16
```

```
plot(lm_weight)
```

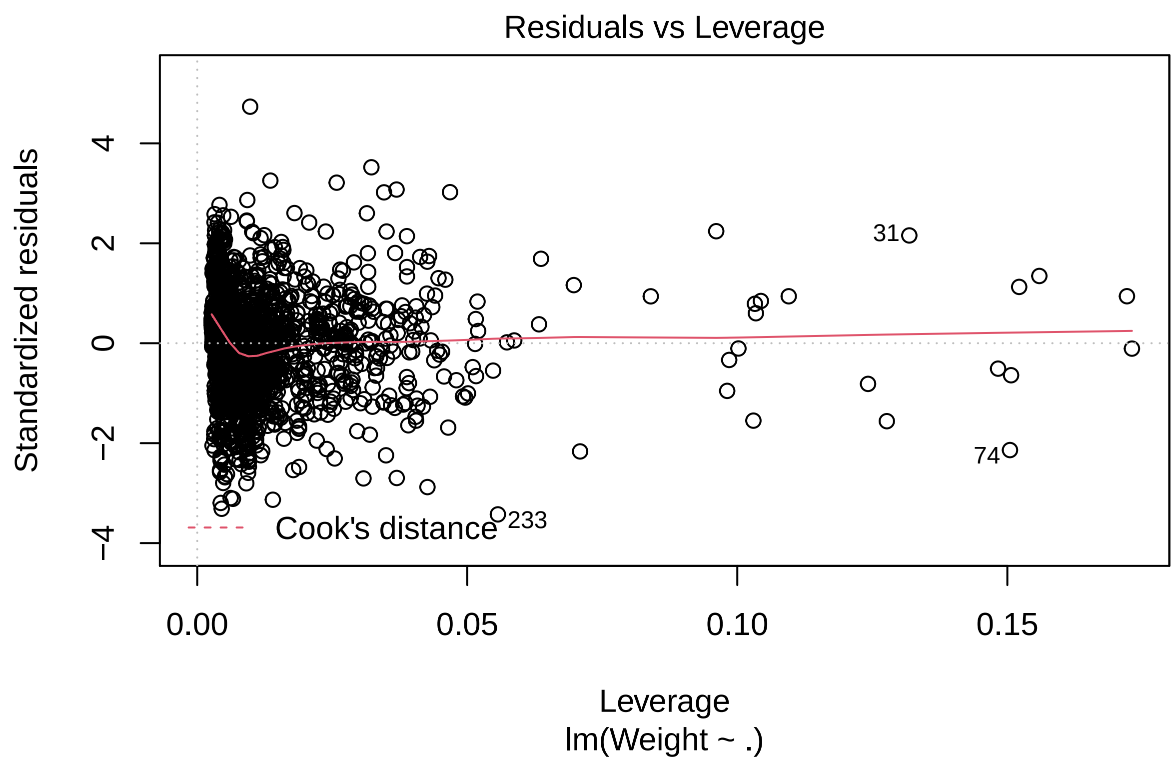
```
## Warning: not plotting observations with leverage one:
##   27
```


Residuals vs Fitted
345
435
233
50
0
−50
Residuals
20
40
60
80
100
120
Fitted values
lm(Weight ~ .)


Normal Q−Q
345
435
233
4
2
0
−2
Standardized residuals
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(Weight ~ .)



Scale−Location
345
435
233
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
20
40
60
80
100
120
Fitted values
lm(Weight ~ .)



Residuals vs Leverage
31
74
233
Cook's distance
Standardized residuals
−4
−2
0
2
4
0.00
0.05
0.10
0.15
Leverage
lm(Weight ~ .)