Data Mining Project (MaBAn 2020)

Predicting obesity levels according to daily habits

by : Ángel Tomás-Ripoll & Laurence Tréteault-Falsafi

Contents

Introduction **1**

Data Pre-Processing **4**

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)
source("VIF.R")

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


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

```r
    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 :

```
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.

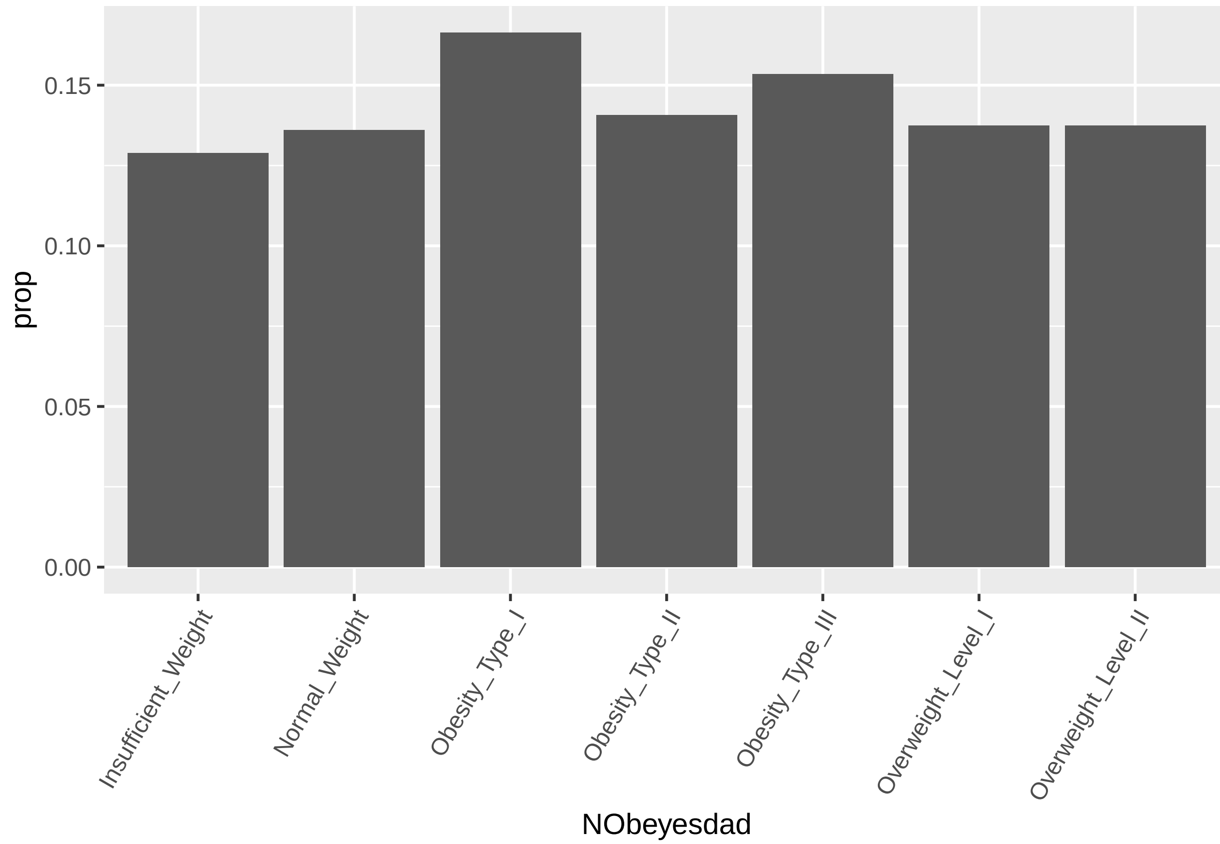
Data Pre-Processing

```
obesity$FCVC[obesity$FCVC <= 1] <- "Never"

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

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

```
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.

```
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_with_overweight	FAVC	FCVC
Length:2111	Length:2111	Length:2111
Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character
NA	NA	NA
NA	NA	NA
NA	NA	NA

Table continues below

NCP	CAEC	SMOKE	CH2O
Min. :1.000	Length:2111	Length:2111	Min. :1.000
1st Qu.:2.659	Class :character	Class :character	1st Qu.:1.585
Median :3.000	Mode :character	Mode :character	Median :2.000
Mean :2.686	NA	NA	Mean :2.008
3rd Qu.:3.000	NA	NA	3rd Qu.:2.477
Max. :4.000	NA	NA	Max. :3.000

Table continues below

SCC	FAF	TUE	CALC
Length:2111	Min. :0.0000	Min. :0.0000	Length:2111
Class :character	1st Qu.:0.1245	1st Qu.:0.0000	Class :character
Mode :character	Median :1.0000	Median :0.6253	Mode :character
NA	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

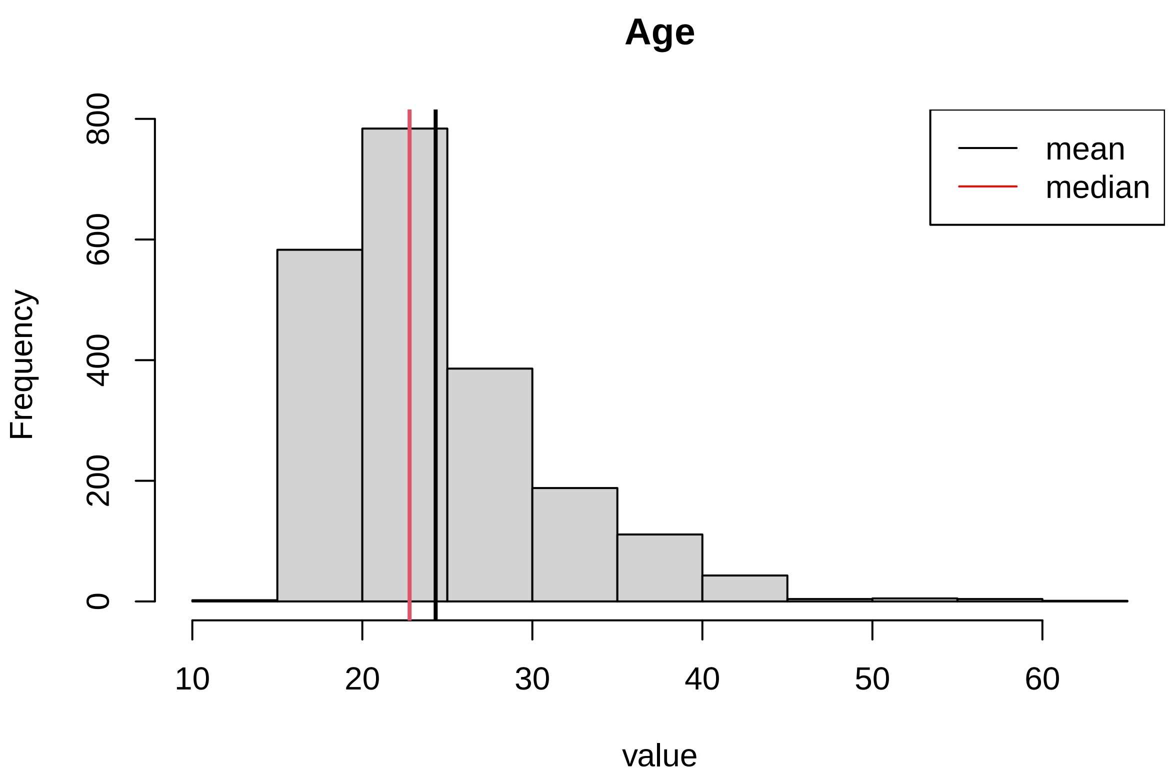
MTRANS	NObeyesdad
Length:2111	Length:2111
Class :character	Class :character
Mode :character	Mode :character
NA	NA
NA	NA
NA	NA

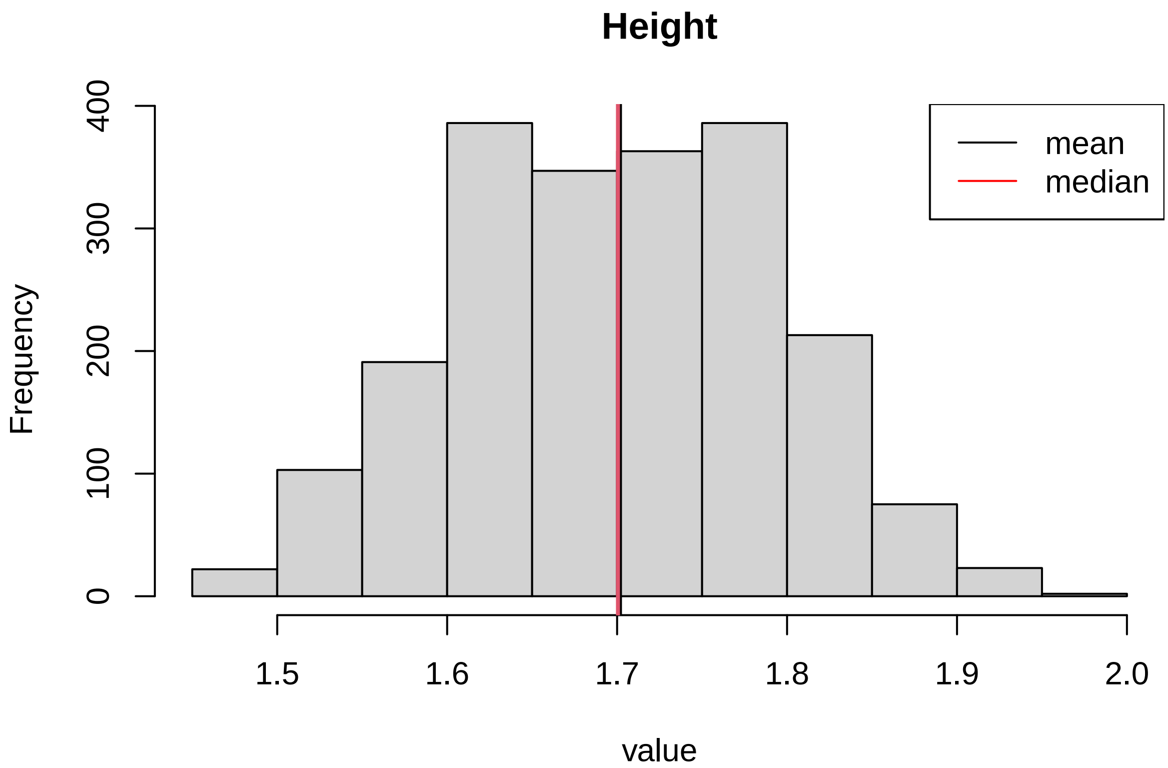
```r
str(obesity)
```

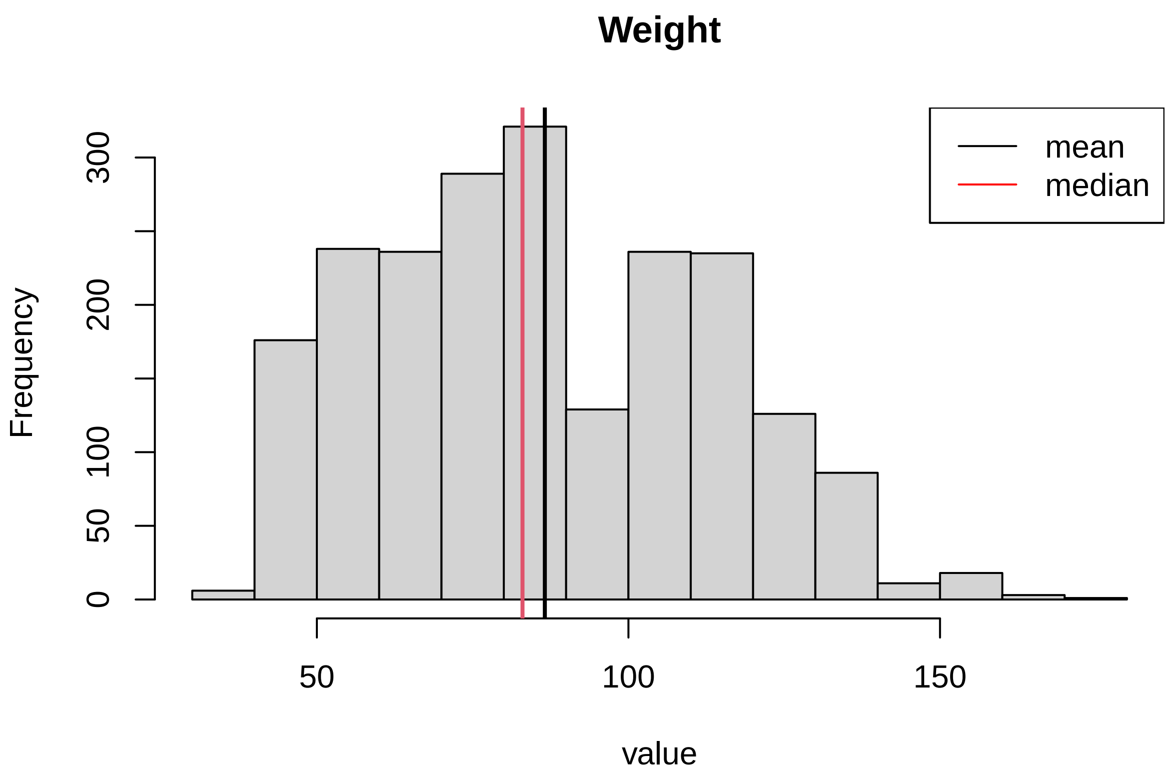
```
## 'data.frame':    2111 obs. of  17 variables:
##  $ Gender                        : chr  "Female" "Female" "Male" "Male" ...
##  $ Age                           : num  21 21 23 27 22 29 23 22 24 22 ...
##  $ Height                        : num  1.62 1.52 1.8 1.8 1.78 1.62 1.5 1.64 1.78 1.72 ...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

```
##  $ FAF                            : num  0 3 2 2 0 0 1 3 1 1 ...
##  $ TUE                            : num  1 0 1 0 0 0 0 0 1 1 ...
##  $ CALC                           : chr  "no" "Sometimes" "Frequently" "Frequently" ...
##  $ MTRANS                         : chr  "Public_Transportation" "Public_Transportation" "Public_Trans
##  $ NObeyesdad                     : chr  "Normal_Weight" "Normal_Weight" "Normal_Weight" "Overweight_I
```

```
for (i in 2:4) {
    hist(obesity[, i], breaks = 10, main = names(obesity[i]),
        xlab = "value", freq = T)
    abline(v = mean(obesity[, i]), col = 1, lwd = 2)
    abline(v = median(obesity[, i]), col = 2,
        lwd = 2)
    legend("topright", legend = c("mean", "median"),
        col = c("black", "red"), lty = 1)
}
```




Height
Frequency
0
100
200
300
400
1.5
1.6
1.7
1.8
1.9
2.0
value
mean
median


Weight
Frequency
0
50
100
200
300
50
100
150
value
mean
median

Now, to have an idea of the distribution of the categorical variables, we will first replace the numbers with a string corresponding to its category, and then implement a `for` loop to get the frequencies.

```
obesity$FCVC[obesity$FCVC <= 1] <- "Never"

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

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



for (i in c(1, 5:10, 12, 15:17)) {
    x = count(obesity, obesity[, i], name = "Count")
    colnames(x)[1] = colnames(obesity[i])
    cat(pandoc.table(as.data.frame(summary(as.factor(obesity[,
        i])))))
    print("\n")


}
```

```
##
## -------------------------------------------
##          summary(as.factor(obesity[,
##                           i]))
## ------------ -----------------------------
##  **Female**               1043
##
##   **Male**                1068
## -------------------------------------------
##
## [1] "\n"
##
## ----------------------------------------
##       summary(as.factor(obesity[,
##                         i]))
## --------- -----------------------------
##  **no**                  385
##
##  **yes**                1726
## ----------------------------------------
##
## [1] "\n"
##
## ----------------------------------------
##       summary(as.factor(obesity[,
##                         i]))
## --------- -----------------------------
##  **no**                  245
##
```

```
##  **yes**              1866
## ---------------------------------------
##
## [1] "\n"
##
## ---------------------------------------------
##             summary(as.factor(obesity[,
##                            i]))
## --------------- -----------------------------
##   **Always**                 1309
##
##    **Never**                  33
##
##  **Sometimes**                769
## ---------------------------------------------
##
## [1] "\n"
##
## --------------------------------------------
##            summary(as.factor(obesity[,
##                           i]))
## -------------- -----------------------------
##     **3**                   1203
##
##     **1**                    199
##
##     **4**                    69
##
##  **1.104642**                 2
##
##  **1.73762**                  2
##
##  **1.894384**                 2
##
##  **2.644692**                 2
##
##  **2.77684**                  2
##
##  **3.559841**                 2
##
##  **3.691226**                 2
##
##  **3.985442**                 2
##
##  **1.000283**                 1
##
##  **1.000414**                 1
##
##  **1.00061**                  1
##
##  **1.001383**                 1
##
##  **1.001542**                 1
##
##  **1.001633**                 1
##
##  **1.005391**                 1
##
```

```
##   **1.009426**               1
##
##   **1.010319**               1
##
##   **1.014916**               1
##
##   **1.015488**               1
##
##   **1.02075**                1
##
##   **1.030416**               1
##
##   **1.032887**               1
##
##   **1.044628**               1
##
##   **1.046144**               1
##
##   **1.047197**               1
##
##   **1.049534**               1
##
##   **1.058123**               1
##
##   **1.060796**               1
##
##   **1.068196**               1
##
##   **1.068443**               1
##
##   **1.073421**               1
##
##   **1.075553**               1
##
##   **1.077331**               1
##
##   **1.07976**                1
##
##   **1.081805**               1
##
##   **1.082304**               1
##
##   **1.08687**                1
##
##   **1.089048**               1
##
##   **1.095223**               1
##
##   **1.097312**               1
##
##   **1.09749**                1
##
##   **1.099151**               1
##
##   **1.101404**               1
##
##   **1.10548**                1
##
```

```
##  **1.105617**              1
##
##  **1.109956**              1
##
##  **1.114564**              1
##
##  **1.116401**              1
##
##  **1.120102**              1
##
##  **1.124977**              1
##
##  **1.130751**              1
##
##  **1.131695**              1
##
##  **1.134042**              1
##
##  **1.134321**              1
##
##  **1.135278**              1
##
##  **1.13715**               1
##
##  **1.139317**              1
##
##  **1.146052**              1
##
##  **1.146794**              1
##
##  **1.152521**              1
##
##  **1.154318**              1
##
##  **1.163666**              1
##
##  **1.169173**              1
##
##  **1.171027**              1
##
##  **1.178708**              1
##
##  **1.193589**              1
##
##  **1.193729**              1
##
##  **1.194815**              1
##
##  **1.198643**              1
##
##  **1.202179**              1
##
##  **1.211606**              1
##
##  **1.213431**              1
##
##  **1.226342**              1
##
```

```
##  **1.231915**               1
##
##  **1.237454**               1
##
##  **1.240046**               1
##
##  **1.240424**               1
##
##  **1.24884**                1
##
##  **1.250548**               1
##
##  **1.25535**                1
##
##  **1.259628**               1
##
##  **1.259803**               1
##
##  **1.262831**               1
##
##  **1.265463**               1
##
##  **1.271624**               1
##
##  **1.273128**               1
##
##   **1.2919**                1
##
##  **1.293342**               1
##
##  **1.296156**               1
##
##  **1.311797**               1
##
##  **1.313403**               1
##
##  **1.317884**               1
##
##  **1.320768**               1
##
##  **1.322087**               1
##
##  **1.326982**               1
##
##  **1.338033**               1
##
##  **(Other)**               536
## --------------------------------------------
##
## [1] "\n"
##
## ----------------------------------------------
##              summary(as.factor(obesity[,
##                              i]))
## ---------------- -----------------------------
##    **Always**                 53
##
##  **Frequently**               242
```

```
##
##      **no**                   51
##
##  **Sometimes**               1765
## ---------------------------------------------
##
## [1] "\n"
##
## --------------------------------------
##       summary(as.factor(obesity[,
##                       i]))
## --------- ----------------------------
##  **no**               2067
##
##  **yes**               44
## --------------------------------------
##
## [1] "\n"
##
## --------------------------------------
##       summary(as.factor(obesity[,
##                       i]))
## --------- ----------------------------
##  **no**               2015
##
##  **yes**               96
## --------------------------------------
##
## [1] "\n"
##
## ---------------------------------------------
##              summary(as.factor(obesity[,
##                              i]))
## ---------------- ----------------------------
##    **Always**                 1
##
##  **Frequently**              70
##
##      **no**                  639
##
##  **Sometimes**               1401
## ---------------------------------------------
##
## [1] "\n"
##
## --------------------------------------------------------
##                         summary(as.factor(obesity[,
##                                         i]))
## --------------------------- ----------------------------
##       **Automobile**                   457
##
##          **Bike**                       7
##
##        **Motorbike**                   11
##
##  **Public_Transportation**             1580
##
##         **Walking**                    56
```

```
## ----------------------------------------------------------
##
## [1] "\n"
##
## --------------------------------------------------------
##                       summary(as.factor(obesity[,
##                                       i]))
## ------------------------- ----------------------------
##  **Insufficient_Weight**              272
##
##     **Normal_Weight**                 287
##
##    **Obesity_Type_I**                 351
##
##    **Obesity_Type_II**                297
##
##   **Obesity_Type_III**                324
##
##  **Overweight_Level_I**               290
##
##  **Overweight_Level_II**              290
## --------------------------------------------------------
##
## [1] "\n"
```