Exercise 2_cleaning_solutions

Alessandra Meddis

Exercise: cleaning data

We consider extract data from Sundby95 survey carried out in Copenhagen 1995 to assess general health of people. You can find the data in: "https://raw.githubusercontent.com/AMeddis/IntrotoR-for-Basic-Statistics/refs/heads/main/data_exercise/sundby_clean.csv"

The data include the variables:

- kon: sex (1/2)
- v75: weight (kg)
- v76: height (cm)
- v17: physical activity (categories 1-4 with 1: most activity)
- v24af: alcohol intake during the last week.

Exercise 0:

Read the data into R and get a summary of its contents.

```
data_raw<-read.csv("https://raw.githubusercontent.com/AMeddis/IntrotoR-for-Basic-Statistics/refs/heads/
```

Exercise 1:

1. Change the name of the columns (in English) so to be easier to understand what they refer to.

```
colnames(data_raw)<-c("physical_activity","alcohol_intake", "sex","weight","height")
```

2. Use the *str()* function on the data. Check the types of the variables, is there something unexpected?

```
str(data_raw)
```

```
## 'data.frame':    1500 obs. of  5 variables:
##  $ physical_activity: int  2 3 3 3 4 1 3 3 4 3 ...
##  $ alcohol_intake   : num  10 2 0 4 6 0 10 1 0 0 ...
##  $ sex              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ weight           : chr  "75" "58" "" "85" ...
##  $ height           : num  170 175 191 179 180 186 190 176 174 168 ...
```

weight is a charcter, this should be a numerical variable

Exercise 2:

Weight should be a numerical variable, but it has been imported as character. Can you guess why? Note that missing values for *weight* are indicated with an empty space " " .

1. Add one new variable *weight_num* to the data set, that is the numerical version of weight. You can use *as.numeric()*. What is the warning message suggesting?

```
data_raw$weight_num<-as.numeric(data_raw$weight)
```

```
## Warning: NAs introduced by coercion
```

The warning message suggests that R could not recognize the number to associate to some of the elements of the vector

2. The function *is.na* gives TRUE if the element of the vector is an NA, FALSE otherwise.

   The function *sum*, when used on a logical vector, is counting how many elements satisfy the specified condition.

   ```
   Run the commands: sum(data_raw$weight=="") and sum(is.na(data_raw$weight_num))
   ```

   What are these two commands providing?

```
sum(data_raw$weight=="")
```

```
## [1] 51
```

```
sum(is.na(data_raw$weight_num))
```

```
## [1] 53
```

It provides how many missing values there are in weight and how many there are in weight_num. We can see that there are 2 more missing elements in weight_num, even if this should be the same number since they are the same variable but of different type.

3. The function *which* provides the index (position) of the elements where the specified condition is TRUE.

   ```
   Run the command: which(data_raw$weight!="" & is.na(data_raw$weight_num)).
   ```

What are these numbers? **Note**: & is the logical operator AND (TRUE where both conditions are fulfilled)

```
 which(data_raw$weight!="" & is.na(data_raw$weight_num))
```

```
## [1] 85 99
```

these two are the indeces of the location of the elements coerced by R

4. Correct manually the weights values in *weight*, re-create the numerical version of weight.

```
data_raw[c(85,99),]$weight
```

```
## [1] "84,5" "79,5"
```

```
data_raw[c(85,99),]$weight<-c(84.5,79.5)

data_raw$weight_num<-as.numeric(data_raw$weight)
```

Exercise 3

Sex (kon) is a numerical variable, with values 1 and 2. We believe that 1 represents males and 2 females (default). However, we would like to check if this assumption is correct.

1. We have the information on the height, calculate the median height by sex, which group is taller? Is the assumption correct?

```
tapply(data_raw$height, data_raw$sex, FUN=median, na.rm=TRUE)
```

```
##   1   2
## 180 168
```

Group for sex=1 is taller, so the assumption is correct

2. Transform sex (kon) into a factor and assign more appropriate labels.

```
data_raw$sex<-factor(data_raw$sex, levels=c(1,2), labels=c("M","F") )
```

3. Is there another variable you would like to have as a factor? If yes, transform it.

```
summary(data_raw)
```

```
##  physical_activity alcohol_intake    sex         weight
##  Min.   :1.000     Min.   : 0.00   M   :647   Length:1500
##  1st Qu.:3.000     1st Qu.: 0.00   F   :827   Class :character
##  Median :3.000     Median : 0.00   NA's: 26   Mode  :character
##  Mean   :2.895     Mean   : 1.62
##  3rd Qu.:3.000     3rd Qu.: 2.00
##  Max.   :4.000     Max.   :30.00
##  NA's   :48        NA's   :57
##      height        weight_num
##  Min.   :148.0   Min.   : 32.0
##  1st Qu.:166.0   1st Qu.: 60.0
##  Median :172.0   Median : 70.0
##  Mean   :172.9   Mean   : 70.9
##  3rd Qu.:180.0   3rd Qu.: 80.0
##  Max.   :200.0   Max.   :140.0
##  NA's   :59      NA's   :51
```

```
data_raw$physical_activity<-factor(data_raw$physical_activity, levels=1:4, labels=c("max","medium","low
```

Exercise 4

We want to calculate the BMI for each patient. The formula is : weight(kg) / height(m) ˆ2, but we have the height in centimeters.

1. Create a new variable *height_m*, which is the height in meters.

```
data_raw$height_m<-data_raw$height/100
```

2. Create a new variable for *BMI*.

```
data_raw$BMI <- data_raw$weight_num/ (data_raw$height_m)^2
```