Introduction to R programming: Data preprocessing

Bernard Silenou[1] and Henrik Schanze[1]

[1]Department of Epidemiology, Helmholtz Centre for Infection Research, Braunschweig, Germany

November 23, 2023

Contents

1 Loading packages and data into R **2**
1.1 Loading packages 2
1.2 Loading data 2

2 Cleaning and transforming data **2**
2.1 Cleaning strings 3

Welcome to this R course on basic data management with R. Before proceeding this course we advise you to first get familiar with R-Studio (or what ever IDE you are using) and the usage of R-markdown-notebooks. These prerequisite are covered in the chapter of the course called "Basic R".

1 Loading packages and data into R

Goals

-
- blabla

We are starting with loading the data we want to work with into R. Data could be stored in different kinds of formats. For the majority of common formats there are simple solutions to import that data.
As an example we want to use a *csv* file which stores data about movies including the name, genre, rating and a lot more. We can import the file with the function `read.csv` and give it the name *raw_data*.

1.1 Loading packages

Before loading a package to your current R session, the package needs to already be installed to your computer. Use the command *install.packages* to install a package and `library` to load a package. The RStudio IDE provides an option to search and install package.

```
install.packages("MASS")
library(MASS)
```

The command install.packages would install the needed package from the default R repository called CRAN. If the package that you wish to install in not on CRAN, you would need to search for the repository hosting the package, download the tar.gz file and install it.

1.2 Loading data

```
setwd("~/Introduction-to-R-programming/lecture_notebooks")
dataMovies = read.csv("./data/movies.csv")
```

`.` in the file path represents the current working directory and can be printed using `getwd()` command.

2 Cleaning and transforming data

Goals

After reading this section, you should ba able to do th following:

- Manipulate categorical variables and stings
- Subset a data
- Transform variables in a data
- Convert data from wide to long formats and back
- Sort data

At this point, your data and packages have been successfully loaded to your R instance. Since raw data often has much noise or errors or outliers, etc., it should be processed thoroughly and carefully before fitting a model to it. Data wrangling is the process of transforming raw data into informative data.
Data cleaning includes identifying outliers, error records and missing values, duplicates records, etc.

2.1 Cleaning strings

Data values can be recorded in a way that R does not understand, for example, a question that requires a TRUE or FALSE response may have been recorded as *Y* or *N*, or *Yes* or *No*.
Example: replace the character variable *drugUse* with the logical value *TRUE* or *FALSE*.

```
characterToLogical <- function(x){
y = rep(NA, length(x))
y[x == "Yes"] = TRUE
y[x == "No"] <- FALSE
return(y)
}
dataMovies$drugUseLogical = (characterToLogical(dataMovies$drugUse))
```

Quick exercise: Create a dummy (indicator) variable call drugUseDummy for drugUse. Code Yes with 1 and No with 0. What is the data type of drugUseDummy?