

Preparing your computer and R

Before *Session 1* Wed 23rd Oct

Please follow the advice below to prepare for the computer exercises during the course. We are assuming you have (or will have) R (<https://cran.r-project.org/>) and Rstudio (<https://rstudio.com/products/rstudio/download/#download>) on your computer. These instructions are simply to set up your work environment so that you are ready to begin: the specific commands needed to do any exercise will be provided with the exercise.

Organisation of Course Material

Create a folder called “course5577” or other suitable name on the desktop of your computer (or elsewhere on the hard disk if you prefer) where you will download the various materials provided for the course. Create four sub-folders of “course5577”, named “lectures”, “exercises”, “data” and “reading”. Prior to the course, all data sets will be available for you to save in the “data” folder, and prior to each session of the course, the instructions and solutions (including code) will be provided.

Once you open R, you should set the working folder to “data”, using the menus or a command such as:

`setwd("c:/Documents/.../course5577/exercises/data")`

You can see your current working directory by typing `getwd()`

Some of the exercises will require you to create and save temporary data sets. We advise that you always name such files with a prefix “temp” (e.g. “tempExercise1.rda”, “tempFramingham.rdata”) so that you can clean up the data folder when the exercise is completed.

A note on R commands, functions and packages

R comes with a wide range of built-in commands, but there are also many user written functions for performing specific tasks. These functions are stored in packages, and only become available when the package is installed and loaded. Briefly, an R package is installed by the command:

`install.packages("<package name>")` and loaded by the command `library("<package name>")` and help for a function can be found by calling `help("function name")` or `?function name`

The R packages used on *Day1* are:

- rmarkdown
- knitr
- haven
- tidyverse
- epitools
- survival
- survminer

So you should install these as follows:

`install.packages(c("rmarkdown", "knitr", "haven", "tidyverse", "epitools", "survival", "survminer"))`

(if you already have some of the packages installed, remove their name from the list and install the rest)

Now you are ready for Day 1. Packages used in later sessions will appear at the top of the code file with the solution. For packages from Github, please follow the instructions on the next page (These are also provided in solutions for each day so this is just for your information):

Installing a package available in Github (<https://github.com/>)

To install a package from GitHub, you must first install the **devtools** package and then run an install command (for example, to install the **SamplingDesignTools** package we use in Exercise 3.2 on Day 3):

```
install.packages("devtools")
devtools::install_github("nyilin/SamplingDesignTools")
```

Other Github packages we will use:

DoublingOfCases from <https://github.com/nyilin/DoublingOfCase> (in Exercise 5.1)

These packages may require some of their dependencies to be updated separately. In particular, please install the latest version of **vctr**s before using **SamplingDesignTools**

Homemade functions

An alternative to using R built-in functions (and sometimes more convenient) is to write your own R functions. For example, if you write and run the following code, you will have a function that calculates OR from the 4 numbers in a 2-by-2-table:

```
CALC_OR <- function(a,b,c,d) {
  ODDS_RATIO <- round((a/b) / (c/d), 3)
  return(ODDS_RATIO)
}
```

The OR is computed by typing, for example, `CALC_OR(a = 230, b = 4885, c = 106, d = 4779)`

Some useful R functions for viewing/manipulating data

Base R

head(data) returns the first part of a data frame in the console

View(data) shows the data in a separate dataview

colnames(data) or **rownames(data)** retrieve the row or column names of a matrix or dataframe

matrix creates a matrix from the given set of values

table creates a contingency table

Tidyverse functions

Tidyverse is a set of packages that work in harmony to make it easy for data analysis from importing, manipulating, analysing, and visualizing results (<https://tidyverse.tidyverse.org/>). Here's just some of the commonly used functions:

mutate() generates a new col or make changes to an existing col

filter() filters rows based on condition given

select() select a range of cols provided

slice_sample() randomly selects rows

|> (or **%>%**) pipe function, which passes the object on the left into functions on the right, makes the code easier to read