MM R Codebook

JB Alberge

2024-05-09

Contents

Chapter 1

About

This is a beginner's guide to MM analysis with R. Please contact me `jalberge<at>broadinstitute<dot>org` for feedback and suggestions.

1.1 Pre-requisites

To achieve this practical work, we will use a collection of R packages called tidyverse, available with R version 4. We recommend using the Rstudio software as a development environment. R is a language and a calculator. Rstudio is a graphical development environment built on top of R. Tidyverse is a collection of R packages commonly used for data science.

1. To download and install `R`, please follow instruction at the Comprehensive R Archive Network: https://cran.r-project.org/

2. To download and install `Rstudio Desktop`, go to https://posit.co/download/rstudio-desktop/ and hit the Download Rstudio Desktop for <MacOS/Linux/Windows>.

3. Once RStudio and R are installed, open RSudio and run the following command to install the `tidyverse` collection of `R` packages.

```r
install.packages("tidyverse")
```

4. Finally, load the `tidyverse` R packages within your current R environment with the command below. If this command works fine, you should be all set to start using R!

```r
library(tidyverse)
```

1.2 Good practices

Here are some good practices to write R code. They will help you write clear and maintainable code.

Comment your code with # statements

```r
a <- choose(5, 2) # computes binomial coefficients
```

Avoid obvious comments

```r
x <- 2
y <- 1
# if x is greather than y, then print x
if (x>y) print(x)
```

```
## [1] 2
```

Group your code by task

```r
# assign values
x <- 2
y <- 1

# compute statements
if (x>y) print(x)
```

```
## [1] 2
```

Use a consistent naming scheme

```r
# good !
sum.two.elements <- function(x, y){
  return(x+y)
}
multiply.two.elements <- function(x, y){
  return(x*y)
}
```

```
# bad !
my.function.bis <- function(x, y){
  return(x+y)
}
test.function <- function(x, y){
  return(x*y)
}
```

- DRY principle: Don’t Repeat Yourself. Automate repetitive tasks. The same piece of code should not be repeated, but reused.
- Limit line length
- Set your working directory where you store all the code and the results setwd(‘C:/Users/JohnDoe/MM_R_Codebook/’) Should you receive any error message from the R console, make sure you’ve read it and googled it carefully before going crazy.

Chapter 2

R basic functions

These examples were adapted from B. Michel's introduction to R.

R is a calculator.

```
A = 1+1
A
```

```
## [1] 2
```

To create an object in R, the syntax is Name.of.the.object.to.create <- instructions :

```
# This is a comment
x <- 1 # Assignment
x
```

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

The online R help is very complete. You can reach it with the command `help()` (also `?`). For example, type `help(sum)` (also `?sum`) in the console to get help about the function `sum`.

A vector is a sequence of data points of the same type. A vector can be created with `c()`. Try the following commands:

```
A <- c(1,2,10)
B <- seq(from=0,to=10,length=2)
C <- seq(from=0,to=1,by=0.1)
D <- 1:10
```

```
A;B;C;D
C[1:3]
C[-3]
3 * B
D + 2:11
```

Matrices can be defined using the function matrix():

```
M <- matrix(1:6, nrow = 2, ncol = 3)
M
dim(M)
length(M)
ncol(M)
nrow(M)
rownames(M)
colnames(M)
M[2,c(1,3)]
M[c(TRUE,FALSE),]
M %*% A
t(M)
```

A data.frame is key quantity for statistics in R. A data.frame is a matrix with each line corresponding to an individual and each column corresponding to a variable measured on the individuals. Each column thus represents a single variable (same type across all individuals).

```
var1<-c("a","b","a","fi","jk")
var2<-c(5,8,9,1,3)
tab<-data.frame(var1,var2)
tab
tab$var2
```

The function `read.table()` read the data of a (text or csv) file and import them into R as a data.frame:

```
MyData <- read.table(file= "(complete) file path",
                     header = TRUE,
                     sep = "\t",
                     row.names = )
```

The file argument is a character string, it can be the name of the file if the file is in the work director.

S4 classes can fit all these data types (vectors, data.frames, lists) in their attributes, which are themselves accessible via an @ symbol (use object@data.frame$var1).

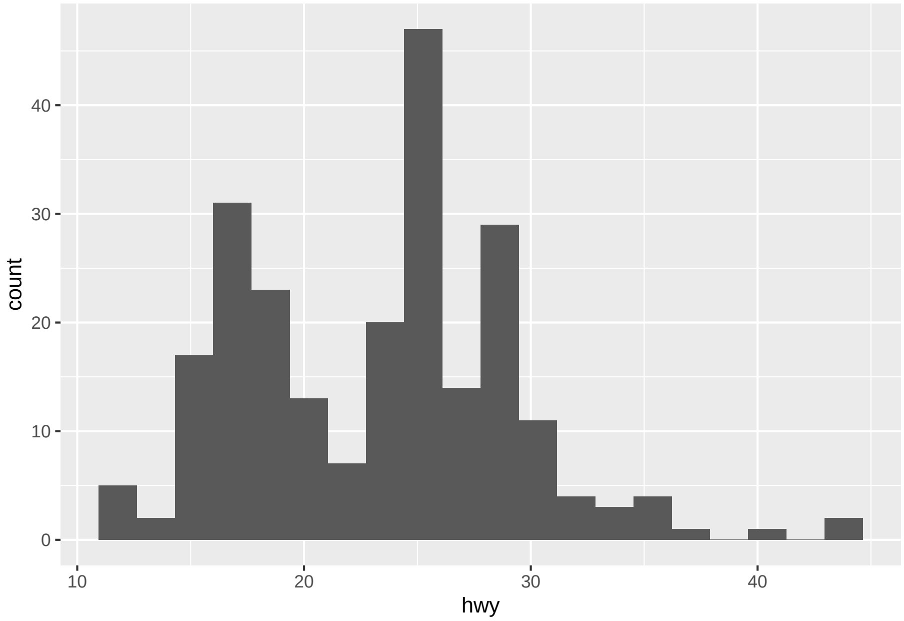
The ggplot2 package, which is part of the tidyverse collection of packages, offers a powerful graphics language for creating elegant and complex plots. It is based on a so-called "Grammar of Graphics" which consists in independently specifying plot building blocks and in combining them to create just about any kind of graphical display you want. Building blocks of a graph include: data, aesthetic mapping (something you can see on a graph), geometric object, statistical transformations, scales…

For this tutorial we only give a few illustrations of ggplot2 graphics. See for instance this pdf cheatsheet for more details.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
data(mpg)
?mpg
data(mpg)
ggplot(mpg) +  aes(hwy) + geom_histogram(bins = 20)
```



Chapter 3

MMRF CoMMpass datasets

This codebook will rely on data from the MMRF CoMMpass data. Register as a researcher and download the following data from release IA21 https://research.themmrf.org/ in a directory named `MMRF_IA21`.

```
CoMMpass_IA21_FlatFile_Dictionaries.tar.gz # decompress this folder
CoMMpass_IA21_FlatFiles.tar.gz # decompress
```

3.1 Explore MMRF CoMMpass clinical variables

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'purrr' was built under R version 4.1.2
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
## Warning: package 'stringr' was built under R version 4.1.2
```

```
## Warning: package 'forcats' was built under R version 4.1.2
```

```
## Warning: package 'lubridate' was built under R version 4.1.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
```