Utilisation de *R* pour l'analyse des données des bases médico-administratives

Séquence 2

Version 2023

L'outil statistique R (www.r-project.org) est devenu incontournable lorsqu'il s'agit d'analyser des données en santé. Libre, gratuit, adossé à une immense communauté d'utilisateurs, ce logiciel permet entre autres de réaliser des analyses statistiques, économiques, spatiales. Il facilite également la génération automatique de rapport d'analyses et la recherche reproductible.

Objectifs

- Prendre en main la programmation sous R
- Prendre en main les nouvelles grammaires pour manipuler les données sous R
- Créer un rapport dynamique avec R

Nolwenn Le Meur, PhD

Enseignant chercheur

Département MéTiS - Méthodes quanTitatives en Santé publique

EHESP | UMR CNRS 6051 | INSERM U1309

Tél: +33 (0) 2 99 02 25 14

Nolwenn.lemeur@ehesp.fr

EHESP

Cartography

Nolwenn Le Meur

May-June 2023

Outline

- data sources and map types
- *ggplot2* and *ggmap* libraries
- *mapsf* library

Data sources

- Spatial data
  - Shape file (ex. IGN website)
  - GeoJSON
  - R libraries
  - Google or OpenStreet map
- Plot it.
  - plot()
  - ggplot2
  - ggmap
  - leaflet
  - mapsf (former cartography package)

Data and map types

- choropleth (rate)
- bubble map (count)
- hexbin
- cartogram
- connection

https://r-graph-gallery.com/index.html

Maps with Google data and ggplot2

spatial data from *maps* library

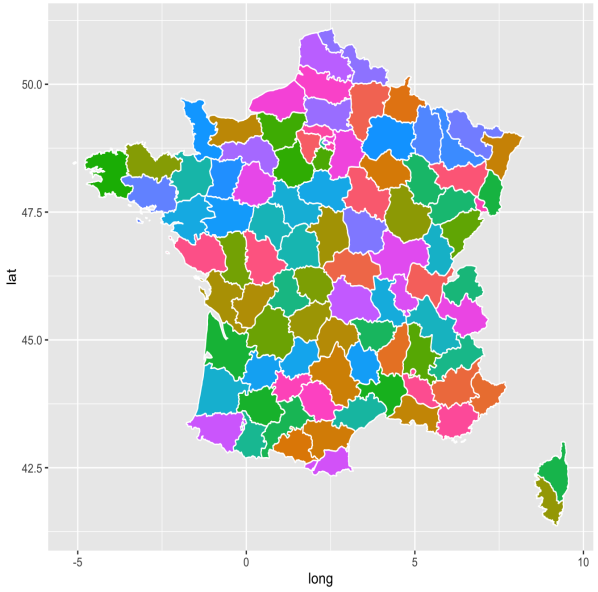
Packages *ggmap* and *maps*

ggmap : A collection of functions to visualize spatial data and models on top of static maps from various online sources (e.g Google Maps and Stamen Maps). It includes functions for geolocation and routing.

```
library(ggmap)
library(maps)
france <- map_data("france")
ggplot(data = france) +
   geom_polygon(aes(x = long, y = lat, fill = region, group = group),
                color = "white") +
   coord_fixed(1.3) +
  guides(fill=FALSE)
```

Maps with Google data and ggplot2

```
## Loading required package: ggplot2
## ℹ Google's Terms of Service: <https://mapsplatform.google.com>
## ℹ Please cite ggmap if you use it! Use `citation("ggmap")` for details.
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Graph gallery examples

https://r-graph-gallery.com/index.html

Maps with the *mapsf* library

https://rgeomatic.hypotheses.org/1729

- To design cartographic representations such as proportional symbols, choropleth, typology, flows or discontinuities maps.
- To improve the graphic presentation of maps, for instance, map palettes, layout elements (scale, north arrow, title…), labels or legends.
- support sf, sp, rgdal, and rgeos geometry data definition.

Read IGN data

https://geoservices.ign.fr/adminexpress

```
#library to import map of France departments
library(sf)
#library(cartography) -deprecated
library(mapsf)

# import ING shape files as an sf object
FrMap <- st_read(dsn="ADE-COG_1-1_SHP_LAMB93_FR/.",
                 layer="DEPARTEMENT", quiet = TRUE)
```

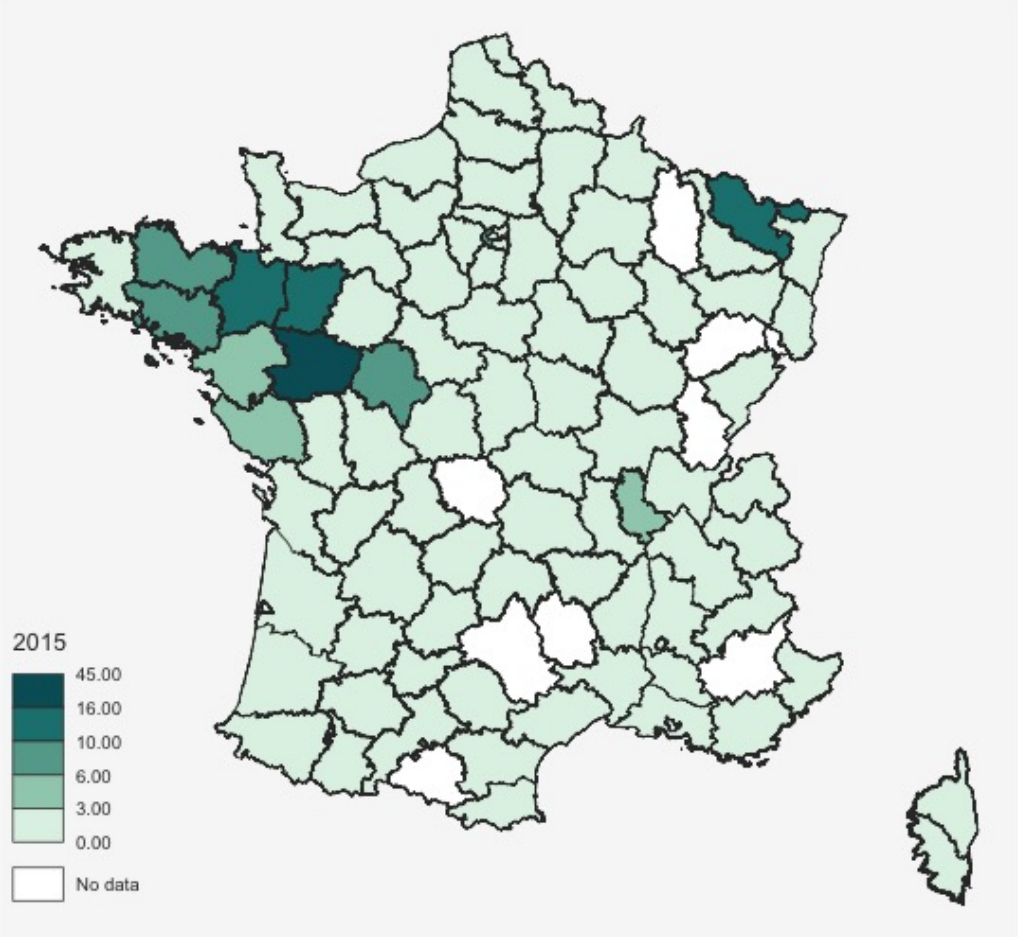
Merging data and sf object

```
Rate <- FrAmbu %>% group_by(DPT, YEAR) %>%
            summarise(Rate_pro=round(sum(outpatient_pro)*100/sum(total_pro),2),
            Rate_stay=round(sum(outpatient_stay)*100/sum(total_stay),2))
Rate_stay_col <- Rate[,-3] %>% spread(YEAR, Rate_stay)

# Merging geo data and pmsi data
FrMap_stay <- full_join( FrMap, Rate_stay_col, by="DPT" )
```

Choropleth map with *mapsf*

```
library(mapsf)
mf_map(FrMap_stay, var = "2015", type = "choro",
       breaks = c(0,3,6,10,16,45))
```

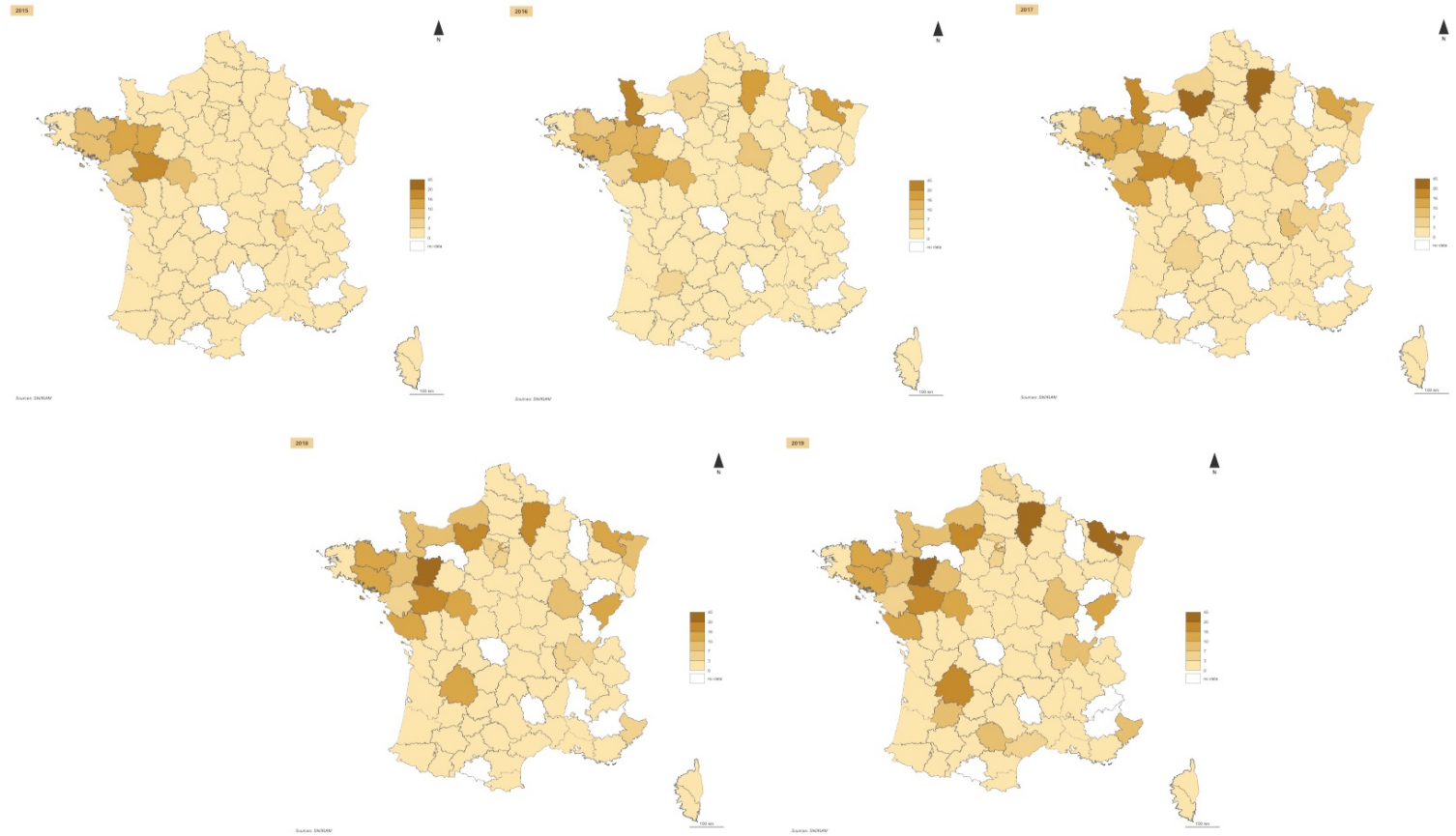


Choropleth map with *mapsf*

```
plot(st_geometry(FrMap_stay), col = NA, border = NA, bg = NA)
# main plot
choroLayer(
  x = FrMap_stay,
  var = "2015",
  breaks = c(0,3,7,10,16,20,45),
  col = carto.pal(pal1 = "sand.pal", n1 = 6),
  border = "grey40",
  lwd = 0.2,
  legend.pos = "right",
  legend.title.txt = "",
  add = TRUE
)
# layout
layoutLayer(title = "2015",
            sources = "Sources: SNIIRAM",
            frame = FALSE, north = FALSE, tabtitle = TRUE, theme= "sand.pal")
# north arrow
north(pos = "topright")
```

Evolution of ambulatory endovascular procedures

Analysis of OpenCCAM data, from 2015-2019.



Le Meur N, Padilla C, Ghoroubi N, Lamirault G, Chatellier G, Gouëffic Y. Geographical Ambulatory Endovascular Revascularisation Disparities in France From 2015 to 2019. European Journal of Vascular and Endovascular Surgery. 2022 Jun;63(6):890–7.

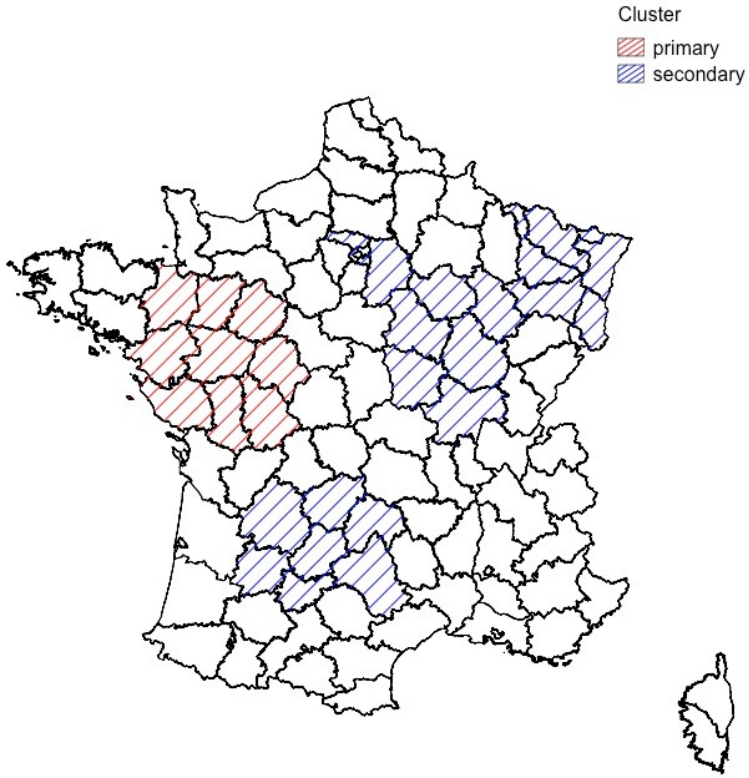
Spatial statistical analysis

Packages :

- *`spdep - descriptive statistics`* :
- create spatial weights matrix objects permitting their use in spatial data analysis
- test for spatial ‘autocorrelation’, including global Moran’s I and Local indicators of spatial association (LISA)
- *`scanstatistics & smerc - clustering analysis`*

Ambulatory endovascular treatments

Analysis of OpenCCAM data, from 2015-2019.



Départements dynamiques (n=27) versus autres départements (n=56) *	OR [95%CI]	p-value
Population		
Proportion de 60 ans et plus	0.50 [0.2-0.8]	0.04
Proportion d'hommes vivant seul de 75 et plus	0.37 [0.2-0.8]	0.01
Pauvreté des 50-59 ans	0.69 [0.5-0.9]	0.02
Tx mortalité maladie artérielles thrombo-emboliques		
Femme	0.36 [0.1-0.8]	0.02
Hommes	3.15 [1.2-8.1]	0.01
Taux de mortalité toutes causes	1.03[1.0-1.1]	<0.01
Offres de soin		
Densité des professionnels de santé (/100 000 habitants):		
kinésitherapeutes	1.02 [1.0-1.1]	0.02
infirmiers	0.98 [0.9-1.0]	0.02

Loop, function and beyond

Nolwenn Le Meur

May-June 2023

Univariate analysis of risk factor of low birthweight

```
library(MASS)
data("birthwt")
head(birthwt, n=3)
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    2     0   0  0  1   0 2523
## 86   0  33 155    3     0   0  0  0   3 2551
## 87   0  20 105    1     1   0  0  0   1 2557
```

Univariate analysis part 1

```
fisher.test(birthwt$low , birthwt$race)
```

Univariate analysis of risk factor of low birthweight

Univariate analysis part 2

```
fisher.test(birthwt$low , birthwt$smoke)
fisher.test(birthwt$low , birthwt$ht)
fisher.test(birthwt$low , birthwt$ui)
```

How to store and display only the p-value of astatistical test?

Most of R function outputs are lists that can be stored in an object

```
lowRace <- fisher.test(birthwt$low , birthwt$race)
str(lowRace)
## List of 4
##  $ p.value    : num 0.0789
##  $ alternative: chr "two.sided"
##  $ method     : chr "Fisher's Exact Test for Count Data"
##  $ data.name  : chr "birthwt$low and birthwt$race"
##  - attr(*, "class")= chr "htest"
```

You can then access the different elements of the list with indexing *[[..]]* or the *$* sign

```
lowRace$p.value
## [1] 0.07888813
```

How to automatically get all the p-values of several test in 1 vector ?

Loop

Using indexing to loop across variables(columns) and perform multiple statistical tests while doingiteration

```
var2test = c("race","smoke","ht","ui")
res = c()
for (i in 1:length(var2test)){
  varname <- var2test[i]
  temp <- fisher.test(birthwt$low, birthwt[ ,varname])
  res[i] <- temp$p.value
  res
}
names(res) = var2test
```

Loop (2)

Keep only the significant ones ?

```
res
##       race      smoke         ht         ui
## 0.07888813 0.03617650 0.05161187 0.02691811
res[res<0.05]
##      smoke         ui
## 0.03617650 0.02691811
```

dplyr library

```
library(dplyr)
birthwt%>%summarise(across(var2test,~ fisher.test(low,.x)$p.value))
##         race     smoke         ht         ui
## 1 0.07888813 0.0361765 0.05161187 0.02691811
```

purr library

The pattern of looping over a vector, doing something to each element and saving the results is socommon that the purrr package provides a family of functions to do it for you. There is one functionfor each type of output:

- map() makes a list.
- map_lgl() makes a logical vector.
- map_int() makes an integer vector.
- map_dbl() makes a double vector.
- map_chr() makes a character vector.

```
library(purrr)
birthwt%>%
 split(. $race)%>%
 # from base R
 map(~lm(bwt~age,data=.))%>%
 map(summary)%>%
 map_dfr(list("coefficients",8))
## # A tibble: 1 × 3
##     `1`   `2`   `3`
##   <dbl> <dbl> <dbl>
## 1 0.106 0.100 0.814
```

Functions

If you start applying again and again the same lines of codes for different but similar data you maywant to generalize and create your own function

```
mean(birthwt$age)
## [1] 23.2381
# if you decompose
sum(birthwt$age)/length(birthwt$age)
## [1] 23.2381
```

```
mean2 <- function(x){
  sum(x)/length(x)
}
mean2(birthwt$age)
## [1] 23.2381
```

Functions: adding arguments

What if NA ? Let's set to NA the first value of the variable age and compute the mean with *mean2()*

```
birthwt$age[1]<-NA
mean2(birthwt$age)
## [1] NA
```

Need to update the function with an *"na.rm"* argument

```
mean2 <- function(x, na.rm=FALSE){
  if(na.rm==TRUE){
    x = x[!is.na(x)]
  }
  sum(x)/length(x)
}
# as na.rm is set to FALSE by default you have to specify it
mean2(birthwt$age, na.rm=TRUE)
## [1] 23.26064
```

Printable summary

Nolwenn Le Meur

May-June 2023

Kable and *KableExtra* packages

```
library(MASS)
data("birthwt")
head(birthwt, n=3)
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    2     0   0  0  1   0 2523
## 86   0  33 155    3     0   0  0  0   3 2551
## 87   0  20 105    1     1   0  0  0   1 2557
```

- With *kableExtra*

```
library(kableExtra)
options("kableExtra.html.bsTable" = T)
dt<- birthwt[1:5, ]
kable(dt)
```

	low	age	lwt	race	smoke	ptl	ht	ui	ftv	bwt
85	0	19	182	2	0	0	0	1	0	2523
86	0	33	155	3	0	0	0	0	3	2551
87	0	20	105	1	1	0	0	0	1	2557
88	0	21	108	1	1	0	0	1	2	2594
89	0	18	107	1	1	0	0	1	0	2600

Style formating

```
dt %>%
  kbl() %>%
  kable_styling()
```

	low	age	lwt	race	smoke	ptl	ht	ui	ftv	bwt
85	0	19	182	2	0	0	0	1	0	2523
86	0	33	155	3	0	0	0	0	3	2551
87	0	20	105	1	1	0	0	0	1	2557
88	0	21	108	1	1	0	0	1	2	2594
89	0	18	107	1	1	0	0	1	0	2600

Booktabs style table

```
dt %>%
  kbl(caption = "Table 1. Belle table") %>%
   kable_classic_2(full_width = F)
```

Table 1. Belle table

	low	age	lwt	race	smoke	ptl	ht	ui	ftv	bwt
85	0	19	182	2	0	0	0	1	0	2523
86	0	33	155	3	0	0	0	0	3	2551
87	0	20	105	1	1	0	0	0	1	2557
88	0	21	108	1	1	0	0	1	2	2594
89	0	18	107	1	1	0	0	1	0	2600

https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html

Descriptive summary with *gtsummary* package

Nice descriptive statistics table

```
library(gtsummary)
table1 <- birthwt %>%
  tbl_summary(include = c(age, race, smoke))
table1
```

Characteristic	N = 189[1]
age	23 (19, 26)
race	
1	96 (51%)
2	26 (14%)
3	67 (35%)
smoke	74 (39%)

[1] Median (IQR); n (%)

Univariate tests with *gtsummary*

```
tbl_summary(birthwt, include = c(age, race, smoke), by = low, missing = "no") %>%
    add_n() %>% # add column with total number of non-missing observations
    add_p() %>% # test for a difference between groups
    modify_header(label = "**Variable**", stat_1="**Yes**", stat_2="**No**") %>%
    modify_spanning_header(all_stat_cols() ~ "**Low birth weight baby**") %>%
    bold_labels()
```

		Low birth weight baby		
Variable	**N**	**Yes**[1]	**No**[1]	**p-value**[2]
age	189	23 (19, 28)	22 (20, 25)	0.2
race	189			0.082
1		73 (56%)	23 (39%)	
2		15 (12%)	11 (19%)	
3		42 (32%)	25 (42%)	
smoke	189	44 (34%)	30 (51%)	0.026

[1] Median (IQR); n (%)

[2] Wilcoxon rank sum test; Pearson's Chi-squared test

Regression model with *gtsummary*

```
mod1 <- glm(low ~ smoke + age + race, birthwt, family = binomial)

tbl_regression(mod1, exponentiate = TRUE)
```

Characteristic	OR[1]	95% CI[1]	p-value
smoke	2.99	1.47, 6.33	0.003
age	0.96	0.90, 1.03	0.3
race	1.70	1.15, 2.55	0.009

[1] OR = Odds Ratio, CI = Confidence Interval

gtsummary, *KableExtra*, and Markdown

Not all output types are supported by the *gtsummary* and KableExtra package

Print Engine	Function	HTML	Word	PDF	RTF
gt	`as_gt()`				
flextable	`as_flex_table()`				
huxtable	`as_hux_table()`				
kableExtra	`as_kable_extra()`				
kable	`as_kable()`				
tibble	`as_tibble()`				

References

https://www.danieldsjoberg.com/gtsummary/

https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html

R Markdown

an authoring framework for data science.

Nolwenn Le Meur

May-June 2023

French School of Public Health

Lecture adapted from https://rmarkdown.rstudio.com/

Objectives

- Connect Data, Methods and Results
- Reproducible research
- Transparency in analytical methodology
- Public availability and reusability of data analysis (if public available data).
- Public accessibility and transparency of scientific communication.

R script versus Rmarkwon report

File name : myscript.R

```
#N. Le Meur
# Nov 5, 2020

#loading data
data(msleep)
#computing summary
summary(msleep)
```

File name : myreport.Rmd

````
---
title: ”Study Report "
author: "Nolwenn Le Meur"
date: "November, 2-6 2020"
output:
  pdf_document: default
---

## Introduction
This studyon reports the variation of sleeping parameters among mammalian species …blablabla

```{r load data, echo=FALSE, eval=TRUE}
data(msleep)
summary(msleep)
```
````

Installation

R Markdown is free and open source package.

```
install.packages("rmarkdown")
```

For nice rendering, you need the *knitr* library

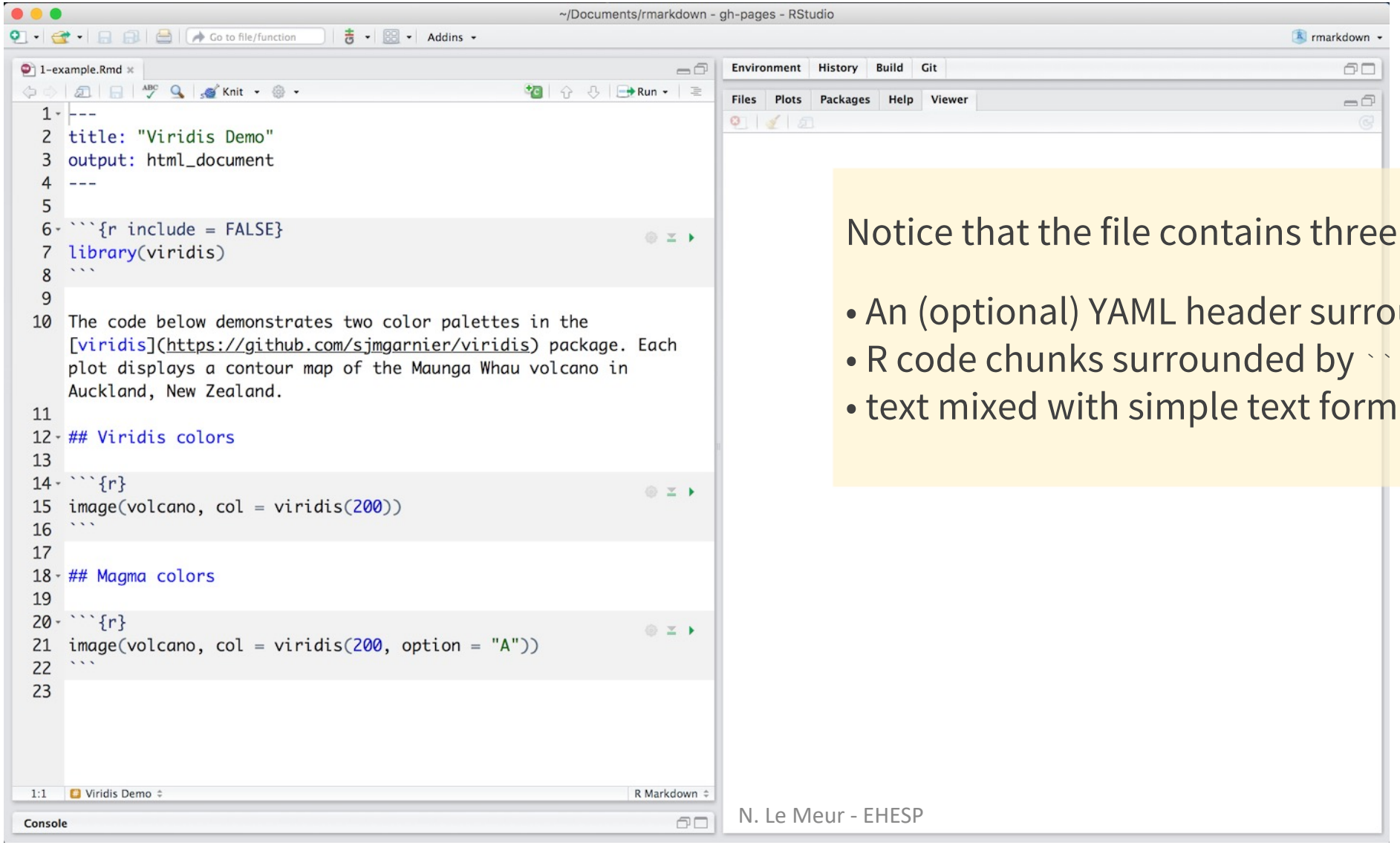
```
install.packages("knitr")
```

And *TinyTex* library for PDF output

```
install.packages("tinytex")
tinytex::install_tinytex()
```

How It Works

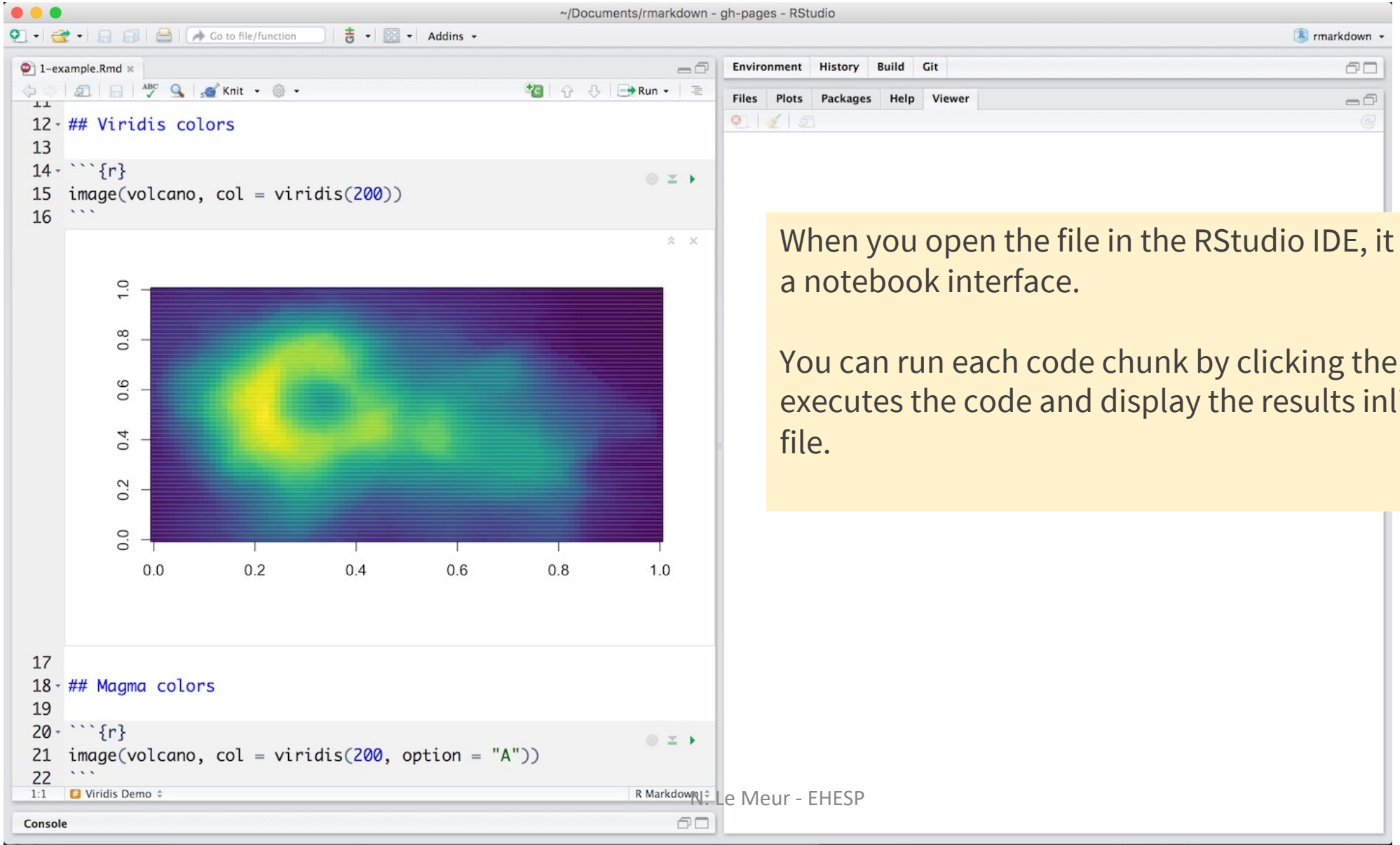
This is an R Markdown file, a plain text file that has the extension `.Rmd`.



Notice that the file contains three types of content:

- An (optional) YAML header surrounded by `---`s
- R code chunks surrounded by ```` ```s ````
- text mixed with simple text formatting

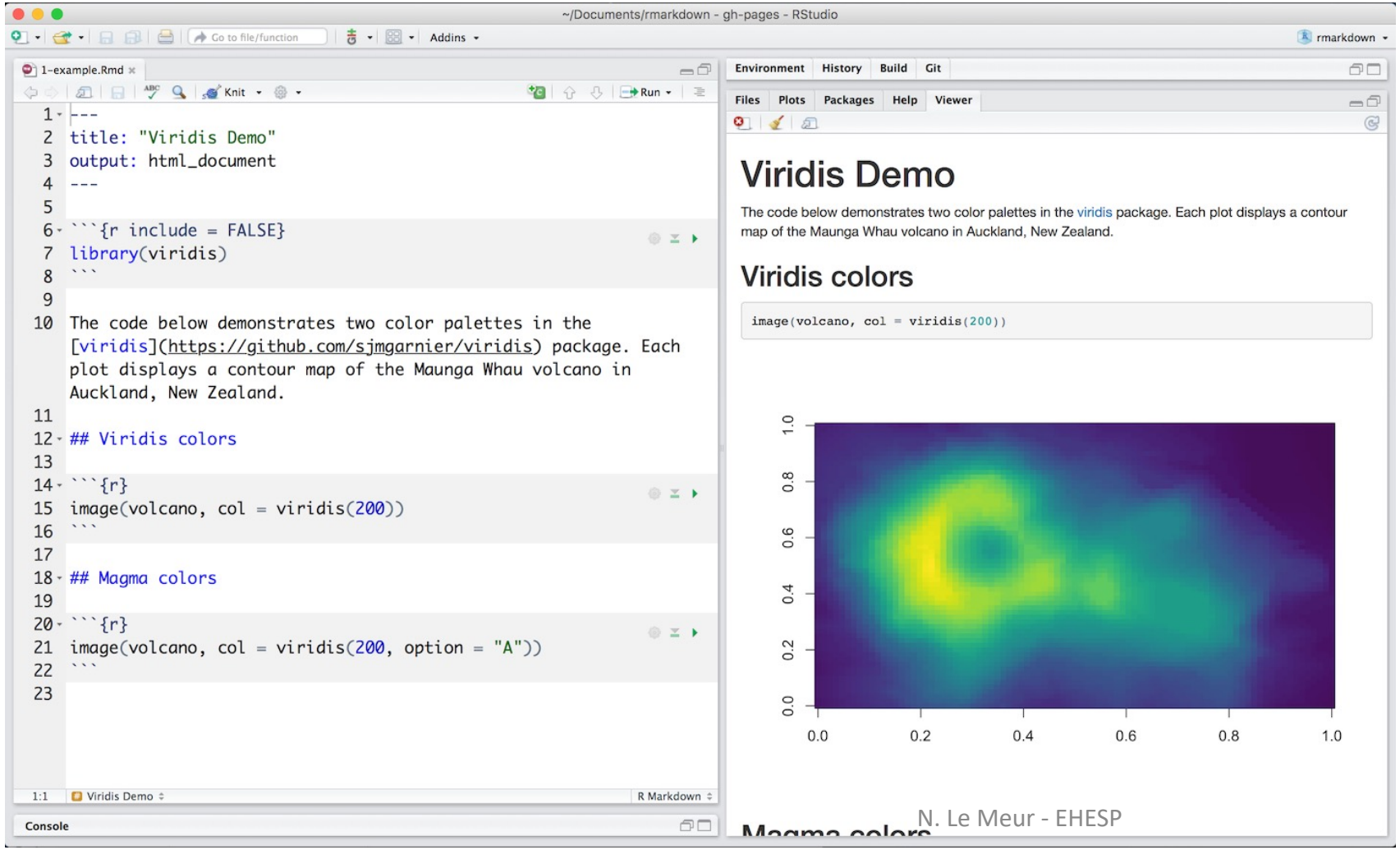
A notebook interface



When you open the file in the RStudio IDE, it becomes a notebook interface.

You can run each code chunk by clicking the icon. RStudio executes the code and display the results inline with your file.

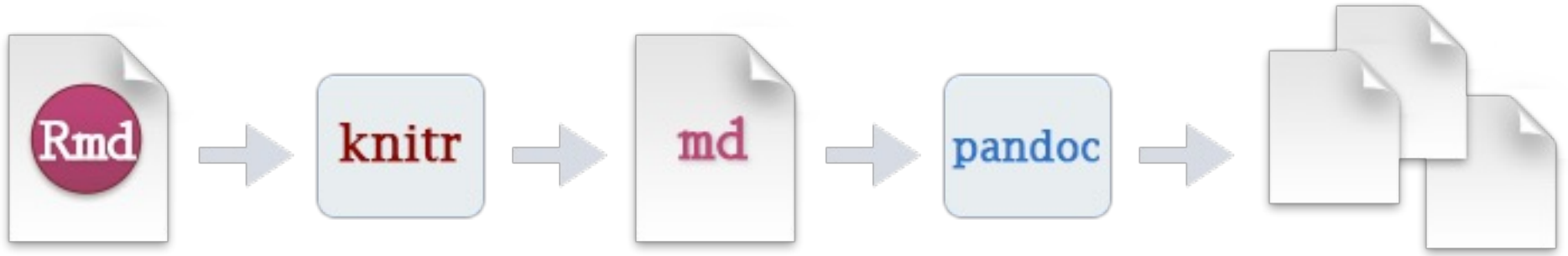
Rendering output



To generate a report from the file, run the “Knit” button in the RStudio IDE to render the file.

R Markdown generates a new file that contains selected text, code, and results from the .Rmd file. The new file can be a finished as a web page, PDF, MS WORD, slide show, handout…

How it works



Executable

R Code chunks

~/Documents/rmarkdown - gh-pages - RStudio

2-chunks.Rmd

```
---
title: "Magma Demo"
output: html_document
---

```{r include = FALSE}
knitr::opts_chunk$set(echo = FALSE)
```

```{r message = FALSE, warning = FALSE}
library(viridis)
```

The code below demonstrates the Magma palette in the [viridis](https://github.com/sjmgarnier/viridis) package. It displays a contour map of the Maunga Whau volcano in Auckland, New Zealand.

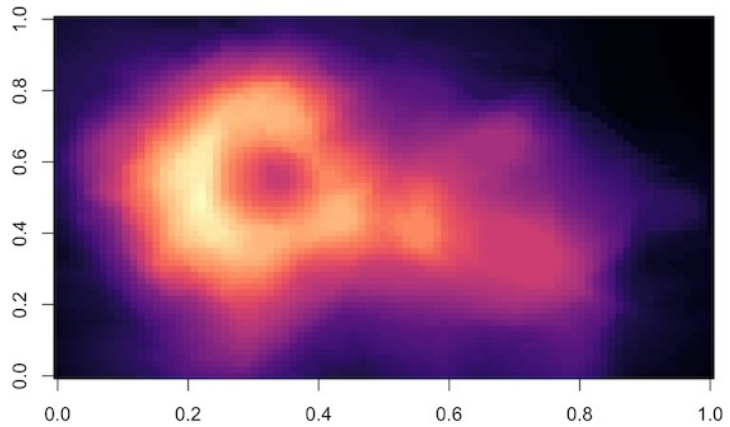
Magma colors

```{r fig.cap = "The Maunga Whau volcano, Auckland."}
image(volcano, col = viridis(200, option = "A"))
```
```

Magma Demo

The code below demonstrates the Magma palette in the viridis package. It displays a contour map of the Maunga Whau volcano in Auckland, New Zealand.

Magma colors



The Maunga Whau volcano, Auckland.

Notice that .Rmd file can also execute code in bash and python…

Chunk Options

```
```{r compute something, echo=FALSE, eval=FALSE, …}

```
```

- Chunk output can be customized with knitr options arguments set in the {} of a chunk header. Above, we use five arguments:
- include = FALSE prevents code and results from appearing in the finished file. R Markdown still runs the code in the chunk, and the results can be used by other chunks.
- echo = FALSE prevents code, but not the results from appearing in the finished file. This is a useful way to embed figures.
- message = FALSE prevents messages that are generated by code from appearing in the finished file.
- warning = FALSE prevents warnings that are generated by code from appearing in the finished.
- fig.cap = "..." adds a caption to graphical results.

See the R Markdown reference guide for a complete list of knitr chunk options

Global Options and caching

- Global options: at the top of the RMD file

To set global options that apply to every chunk in your file, call knitr::opts_chunk$set in a code chunk. Knitr will treat each option that you pass to knitr::opts_chunk$set as a global default that can be overwritten in individual chunk headers.

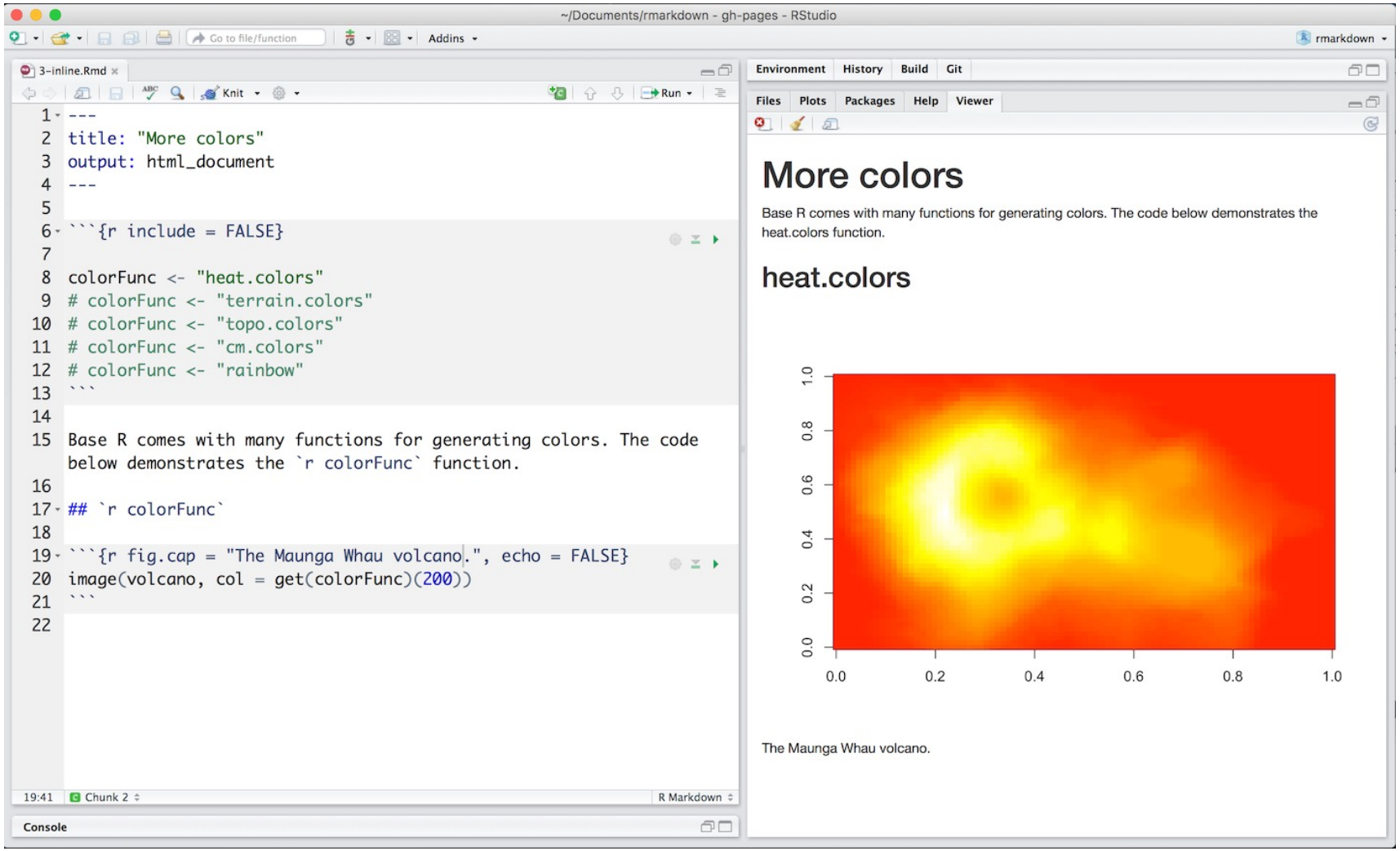
- Caching

If document rendering becomes time consuming due to long computations you can use knitr caching to improve performance.

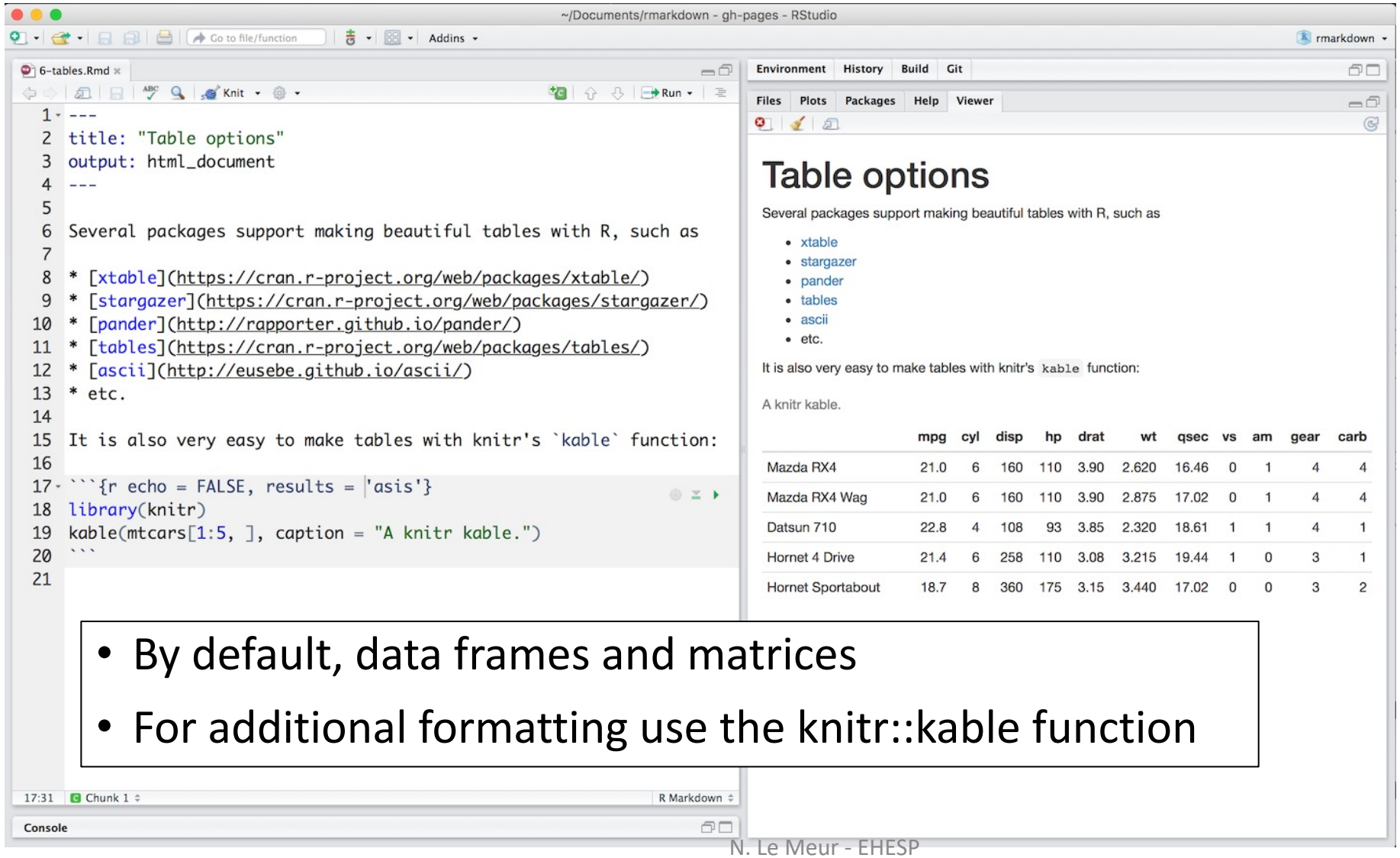
(caching: running chunk code one time and unless the chunk code is updated its result, stored in the cache memory of the computer, is reused)

Inline code

Code results can be inserted directly into the text of a .Rmd file by enclosing the code with `` `r ` ``.

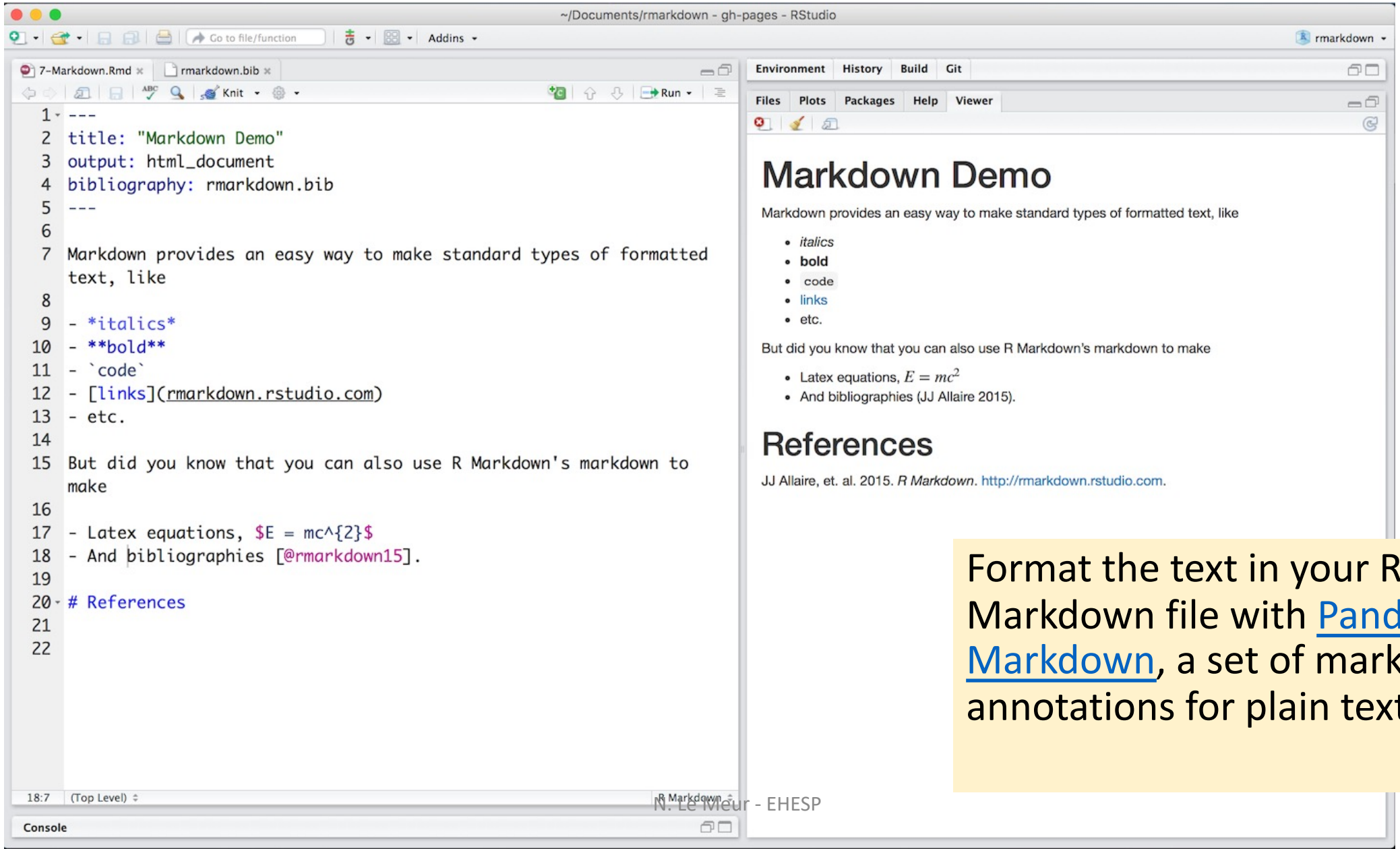


Table



- By default, data frames and matrices
- For additional formatting use the knitr::kable function

Markdown Basics



Format the text in your R Markdown file with Pandoc's Markdown, a set of markup annotations for plain text files.

Ressources

- https://bookdown.org/yihui/rmarkdown-cookbook/

- via websites and interactive documents see https://rmarkdown.rstudio.com/flexdashboard/

For more visits:

- https://rmarkdown.rstudio.com/
- http://larmarange.github.io/analyse-R/rmarkdown-les-rapports-automatises.html