Creating reproductible documents

Luis Miguel Montilla, Alfredo Ascanio

2021/05/19

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

plot(cars)



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Notice that you can render the document in the *Viewer* pane, or in a full window.

# Format

Markdown is a lightweight markup language. This means that we apply format to text using a set of characters, making the process easy to perform on any text editor.

## Headings

We can use # before the text to indicate it is a title

# This is heading  
## Subheading  
### Smaller  
###### Up to six levels

# This is heading

## Subheading

### Smaller

###### Up to six levels

Compare this with regular font

## Emphasis

We can use \* and \_ to bold and italicize text

\*\*bold text\*\*, \*italics\*, \*\*\*both together\*\*\*

I just love **bold text**, sometimes *italics*, but rarely ***both together***

\_one underscore\_, \_\_two\_\_, and even \_\_\_three\_\_\_

We can also use *one underscore*, **two**, and even ***three***

Bold and Italic Best Practices

Markdown applications don’t agree on how to handle underscores in the middle of a word. For compatibility, use asterisks to bold and italicize the middle of a word for emphasis.

> We can quote text  
>  
> --- Luis

We can quote text

— Luis

# Equations

We can also add nicely formatted equations using the $ symbol.

$2+2$  
  
The regression had a $R^{2} = 0.85$  
  
$\text{We can add text inside the equation: } A \times B$

The regression had a

# Structure for Reproducible Analysis in Rmd

We can use markdown notebooks to build a reproducible analysis. The idea behind this compendium is preparing a more or less standardized set of files and directories that can be executed for any third-party so they will be able to obtain the same results. The suggested minimal structure:

* **Project directory**: The main project directory should contain an **.Rproj** file and the **R markdown notebook**.
  + **data**: subdirectory where the data is located
  + **figures**: subdirectory where the output figures are located

In general terms, the structure must contain **input** and **output** folders. In the previous example, these folders are represented by *data* and *figures*, but our output in other cases may be more than only figures.

Another extra folder for **analysis** or **scripts** can be added, containing solely R, Python, or other languages’ scripts that are required to run the process.

The **R markdown notebook** (or **.Rmd file**) will contain the pipeline of analysis, including: - Loading the libraries required for analysis - Loading the external functions, if an analysis or scripts folder exist - Loading the data, contained in the input or data folders - Performing the analysis itself - Creating the output, displaying it and/or saving the results in the output or figures folder

Regardless, this framework for reproducible research is supposed to be versatile and adaptable to the specifics of our project. Don’t be afraid to change it in a way that makes sense to you, while keeping it organized for other people to run the process.

Further reading on the topic of Rmarkdown and all its functions can be found at: <https://bookdown.org/yihui/rmarkdown/>

Creating reproductible documents - Using R code

LMM, AA

2021/05/19

Now, we will proceed to make an exercise, keeping in mind what we saw in the previous .Rmd file.

The exercise will consist of the following structure,

* **Project directory**: The main project directory should contain an **.Rproj** file and the **R markdown notebook**.
  + **data**: subdirectory where the data is located
  + **figures**: subdirectory where the output figures are locatedz

And we will guide you step by step

## 1. Create a new R project in a new directory

We can do this within R studio. If you need a refresher, follow these steps:

1. Open a new Rstudio session, so you can keep reading these instructions.
2. Click on file
3. Click on New Project
4. In the Create Project window, click on New Directory
5. In the Project Type window, click on New Project
6. In the Create New Project window:
   * Directory name: *RMarkdown\_Exercises*
   * Create project as a subdirectory of: *c:/…/Rmarkdown\_CourseFolder*

## 2. Create the folder structure

Remember that we will have two folders:

* **data**: containing the datasets provided by us with the course folder
* **figures**: an empty folder in which we will put the output of our analysis

## 3. Create a new R Notebook

Under the file tab in Rstudio, you will see a white page with a green circle and a + sign. If you click on it, the options for R notebook and R Markdown files will appear. Functionally, both files will have the .Rmd extension, and the output of both will be the same. The main difference is the way things are executed in each.

In summary:

1. Create an R Notebook
2. Delete the content **below** the first four lines:

---  
title: "R Notebook"  
output: html\_notebook  
---

1. You can add more lines to the previous chunk of codes within the dashes *—*, such as author, date, and other information you may find relevant

---  
title: "R Notebook"  
author: "Student Name"  
date: "2021/08/04"  
output: html\_notebook  
---

The *output* line is something Rstudio uses to define which type of document needs to be created at the end of the process. It may be an HTML (like a webpage), a PDF, or a Word document. For example, the document you are reading right now (in any of the three formats mentioned before) was created using RMarkdown.

Once we verify that the structure we need for our analysis is complete, we will proceed with the next steps.

# 4. Add a brief explanation or introduction to your .Rmd file

In lieu of the content we just deleted, we can start by adding a brief (or in-depth) explanation of what we are going to do. Remember that even if introductions are the first thing we read, they can be the last thing to be written or rechecked. So feel free to add something brief for now and later on you can add a more extensive introduction on what we did.

This will be a brief exercise on data managing and data visualization to explore an ecological pattern on the turtle *Chrysemys picta*. Using data from the paper:

Lisa E. Schwanz, Rachel M. Bowden, Ricky-John Spencer, and Fredric J. Janzen. 2009. Nesting ecology and offspring recruitment in a long-lived turtle. Ecology 90:1709.

We will explore graphically two questions:

1. Did the number of nest studied change by year?
2. Was there a difference in clutch size between the years covered by the study?

# 5. Create a chunk to load the packages we will use for this.

Remember that: \* To add a chunk you can click on *Insert Chunk*, or press *Ctrl+Shift+Enter* \* To load an R library, we can use the functions *library(packageName)* or *require(packageName)* \* To install a package, we can use the function *install.packages(“packageName”)*

The packages we will load are: \* tidyverse

You can add some text before or after the chunk to explain what this is doing

# 6. Create a new section in your R Notebook for loading and cleaning data

Remember that your dataset is located in a specific folder within the R project we created before. Create a new R chunk and read your data into R. You can name it however you want, but we suggest the name *turtles* as a standard moving forward in the exercise.

1. Create a chunk to load the data into R
2. Create a new chunk for data cleaning

* Print the first and last 6 rows of your data using the *head()* and *tail()* functions

1. In the same chunk, or a new one, let’s remove the empty rows that were present at the end of our dataset. We can use the function complete.cases(turtles) to subset this.

After this, only rows without NAs should be present. How you perform this filtering is up to you. You can use base R filtering, dplyr filtering, or any other way you feel comfortable with, even going to the .txt file and removing those rows by hand. We will try to avoid the latter option in particular, as we prefer to keep the raw data as it is for transparency. The code will serve as the steps and guidelines for analysis replicability, even regarding data cleaning.

# 7. Create a new section within your R notebook for Data Analysis, a new subsection for Question 1 and a new set of chunks

Question 1 says: **Did the number of nest studied change by year?**

Think for a second how would you approach this question numerically and then graphically? We are not talking necessarily about an hypothesis test, we just want to **see a possible trend** in the number of nest evaluated per year.

Without going further in the paper, a quick exploratory analysis like this may tell us whether or not the experimenters decided to use a standard number of nests per year in the site of their study, or if the number of nests varied across years. If the latter is true, we may ask ourselves why. Maybe the researchers tried to keep a standard number, but some years they encountered less than that. Or maybe they tried to find as many as possible within their allotted experiment time per year. If this is the case, we may be able to correlate the number of nests encountered to other environmental or biological variables available for the site on those years. Some of these variables may be climatic variables, others may be vegetation or landscape change, etc.

You may have an answer by now, and we encourage you to go with it. Now we will propose our way to do it. Remember that there can be more than one path to an appropriate result.

We suggest to generate a frequency table by year, and then a barplot where the height of the bars represent the number of nests per year. We can discuss and add other different options you have in mind.

**Steps to follow:** 1. Create the table of frequency per Year on the *turtles* dataframe 2. Convert this table into a data frame 3. Make your plot 4. Save your plot into the *Figures* folder

If you struggle with how to follow these steps, we will provide you with guidance on the code

Describe your figure. What do you see? What do you think may have happened? You can check the paper to see if there is an explanation given by the authors.

# 8. Create a new subsection for Question 2 and a new set of chunks

Again, think for a second how would you address this question, both mathematically and graphically. We think that this question can potentially have some statistics to it, but let’s omit that part for now.

For this, we suggest the use of boxplots per year, showing the distribution of clutch sizes.

For this, there may not be any necessary data cleaning step.

**Steps to follow:** 1. Clean the data frame based on the Clutch\_Size column. Missing values are recorded as -999.9. Remove these rows 2. Create a boxplot 2. Save the plot into the *Figures* folder

## Exercise conclusion

You have created your (maybe) first Rmarkdown notebook to perform an exploratory data analysis. What are the advantages of this?

1. **Replicability and shareability:** You can send this folder with the notebook to someone else, and they can perform the exact same analysis as you did, seeing your comments and your thought process.
2. **Changeability:** Let’s imagine that you receive a similar dataset for the same turtle, but from a different site. Using R, and Rmarkdown, will allow you to replicate the same analysis and generate a quick and elegant report, **by just changing a line of code!**. You only need to change what the input data is, **if the structure is the same**
3. **Transparency:** You are allowing people to see the analysis and data in full, to audit your process and offer direct feedback and improvements.
4. **Combining multiple languages:** Some times you will need to perform multiple analyses that may not be all part of any R packages or Python libraries. Thus, you may need to shift from one language to another to be able to perform all the analyses you wish. This is very common in sciences. There are software for very specific steps, depending on the field of study. **RMarkdown allows for multiple language integration.** This is something we will discuss in the next exercise.

Creating reproductible documents - Integrating R and Python code

LMM, AA

2021/05/19

Now, we will proceed to replicate the same exercise we did before, but mixing R and Python code. Python code will serve to read and clean the data, while R code will be used to plot and save the figures.

This is to show you with a simple example how we can use the capabilities of both languages. As you advance in your studies or research, this may be a useful tool to have in your toolbox.

We will skip step 1, 2 and 5 of the previous exercise. We will use the same data and folder structure we had before.

## 3. Create a new R Notebook

Following the same instructions as before, we will create a second R Notebook, and we will answer the same two questions as before, regarding the turtle *Chrysemys picta*, and using data from the paper:

Lisa E. Schwanz, Rachel M. Bowden, Ricky-John Spencer, and Fredric J. Janzen. 2009. Nesting ecology and offspring recruitment in a long-lived turtle. Ecology 90:1709.

* Did the number of nest studied change by year?
* Was there a difference in clutch size between the years covered by the study?

# 5. Create a chunk to load the packages we will use for this.

Remember that: \* To add an **R chunk** you can click on *Insert Chunk*, or press *Ctrl+Shift+Enter*. We will now make distinctions between R and Python chunks. \* To load an R library, we can use the functions *library(packageName)* or *require(packageName)* \* To install a package, we can use the function *install.packages(“packageName”)*

The R packages we will load are: \* tidyverse \* reticulate

We are adding the R package [reticulate](https://rstudio.github.io/reticulate/). Aside from R/Python puns regarding the Reticulated Python ( *Malayopython reticulatus* ), this R package will allow us to use python libraries and functions within our RMarkdown code. Installing this package may take a while.

After reticulate is installed and loaded into R, **we need to install the *pandas* python library**. We can run the following code on console *py\_install(“pandas”)*, or you can leave a record of it in your notebook. Whichever you prefer.

# 6. Create a new section in your R Notebook for loading and cleaning data

Remember that your dataset is located in a specific folder within the R project we created before. Create a new R chunk and read your data into R. You can name it however you want, but we suggest the name *turtles* as a standard moving forward in the exercise.

* Create a new **Python chunk** to load the data into R It should look like this, removing the quotes

"```{python}  
  
```"

Notice that R and python chunks have the same initial structure, and what changes is what’s written between the curly brackets. When you click on *Insert*, you will see other languages aside from R and Python that can also be integrated into an RMarkdown file.

Your first python chunk should load the *pandas* library, and perform the reading of the data frame with its read\_csv function. The code should look like this at the end:

import pandas  
turtles = pandas.read\_csv("Data\data.txt", sep = "\t")  
turtles

1. Create a new chunk for data cleaning
2. Print the *turtles* data frame in the python chunk. It will show you the first and last 5 rows.
3. In the same chunk, or a new one, let’s remove the empty rows that were present at the end of our dataset. We can use the pandas method dropna(). After that, print turtles again. The code within your python chunk should look like this:

turtles = turtles.dropna()  
turtles

After this, only rows without NAs should be present. You may know or learn better ways to perform this filtering in python.

# 7. Create a new section within your R notebook for Data Analysis, a new subsection for Question 1 and a new set of chunks

Question 1 says: **Did the number of nest studied change by year?**

Think for a second how would you approach this question numerically and then graphically? We are not talking necessarily about an hypothesis test, we just want to **see a possible trend** in the number of nest evaluated per year.

Without going further in the paper, a quick exploratory analysis like this may tell us whether or not the experimenters decided to use a standard number of nests per year in the site of their study, or if the number of nests varied across years. If the latter is true, we may ask ourselves why. Maybe the researchers tried to keep a standard number, but some years they encountered less than that. Or maybe they tried to find as many as possible within their allotted experiment time per year. If this is the case, we may be able to correlate the number of nests encountered to other environmental or biological variables available for the site on those years. Some of these variables may be climatic variables, others may be vegetation or landscape change, etc.

You may have an answer by now, and we encourage you to go with it. Now we will propose our way to do it. Remember that there can be more than one path to an appropriate result.

We suggest to generate a frequency table by year, and then a barplot where the height of the bars represent the number of nests per year. We can discuss and add other different options you have in mind.

**Steps to follow in a python chunk:** 1. Create the table of frequency per Year on the *turtles* dataframe 2. Convert this table into a data frame \* One example of these two steps in python code is:

pandas.crosstab(index=turtles['Year'], columns='count')  
  
nestyears = pandas.crosstab(index=turtles['Year'], columns='count')

We created a new object called nestyears, that contains the frequency table

**Steps to follow in an R chunk:** 3. Make your plot 4. Save your plot into the *Figures* folder

For these two steps, we already created the frequency table in python. All python objects we have loaded and created are stored within an object called ***py*** in our R session. We can extract these objects and put them in R chunks by using the following statement **py$ObjectName**. As such, your code to make the plot in an R chunk, using the data we have manipulated with python, should look like this:

py$nestyears %>%  
 ggplot(aes(x = rownames(.), y = count)) +  
 geom\_bar(stat = 'identity') +  
 theme\_bw() +  
 xlab("Year") +  
 ylab("Nests")

Modify the following chunks accordingly.

# 8. Create a new subsection for Question 2 and a new set of chunks

Again, think for a second how would you address this question, both mathematically and graphically. We think that this question can potentially have some statistics to it, but let’s omit that part for now.

For this, we suggest the use of boxplots per year, showing the distribution of clutch sizes.

For this, there may not be any necessary data cleaning step.

**Steps to follow in a python chunk:** 1. Clean the data frame based on the Clutch\_Size column. Missing values are recorded as -999.9. Remove these rows

Filtering data in python (with the pandas library) can be very familiar to those used to filtering data in base R. A code for this step could look like this:

filtered\_turtles = turtles[turtles['Clutch\_Size'] != -999.9]

In here, we created a new python object called *filtered\_turtles*. Modify the following chunks accordingly.

**Steps to follow in an R chunk:** 2. Create a boxplot 2. Save the plot into the *Figures* folder

## Exercise conclusion

You have created your second Rmarkdown notebook to perform an exploratory data analysis, and now you were able to combine two different languages!