R Markdown

Dr Ebin Deni Raj

02/09/2020

Contents

Data Analysis using R Markdown

Overview

This tutorial is an introduction to data analysis reporting, that is, combining the computer code needed to organize and analyze a dataset, along with comments and explanations to be read and understood by humans. The executable computer code (here, in the R environment) is an integral part of a document that explains what the analyst did, why, and what was discovered. This is one aspect of so-called reproducible research, for which R is an ideal component.

R Markdown provides an authoring framework for data science. You can use a single R Markdown file to both save and execute code generate high quality reports that can be shared with an audience R Markdown documents are fully reproducible and support dozens of static and dynamic output formats. After any data analysis, the expected output is a report. R markdown is the best way to generate reports in R

It can be called as an authoring framework for data science.

The advantages of this approach are several:
1. Every processing step is transparent;
2. Anyone else can repeat the analysis, if they are given access to the same data;
3. Analysis can easily be expanded or adapted;
4. The results of the analysis are generated with the document, so they are by definition synchronized;
5. The analyst's motivations and interpretations are in the same place where the results of the analysis are presented.

R Markdown provides an authoring framework for data science. You can use a single R Markdown file to both

- save and execute code, and generate high quality reports that can be shared with an audience.
- R Markdown was designed for easier reproducibility, since both the computing code and narratives are in the same document, and results are automatically generated from the source code. R Markdown supports dozens of static and dynamic/interactive output formats.

Installation

Like the rest of R, R Markdown is free and open source. You can install the R Markdown package from CRAN with:

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

The steps in using R Markdown in RStudio are:

1. Create a new text document to hold both the R source code and text comments; when saved this has extension .Rmd.

2. Write text to explain your analysis.

3. Between the text add code chunks to hold the R source code that carries out the analysis;
4. As you add code, you can execute it by sending it from the RMarkdown code chunk to the R console.

5. Compile a HTML document from the R Markdown source file, using the Knitr program within RStudio.

6. Open the HTML document in a browser; this is your analysis.

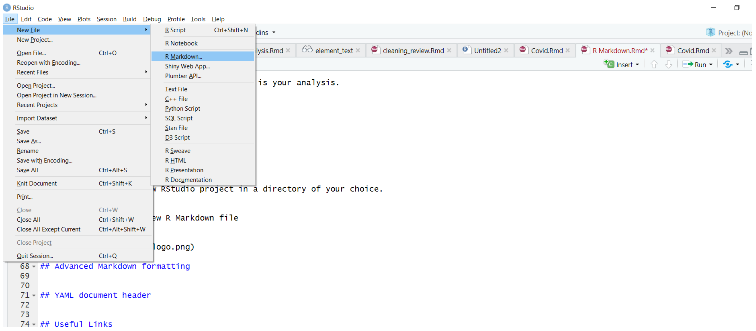
We now go through these steps.

Creating a new R Markdown

Task 1 : Start RStudio.

Task 2 : Start a new RStudio project in a directory of your choice.

Task 3 : Create a new R Markdown file



```
is your analysis.

v RStudio project in a directory of your choice.

ew R Markdown file

logo.png)
## Advanced Markdown formatting


## YAML document header


## Useful Links
```

Figure 1: Create R Markdown file in R studio

Task 4 : Name the new file.

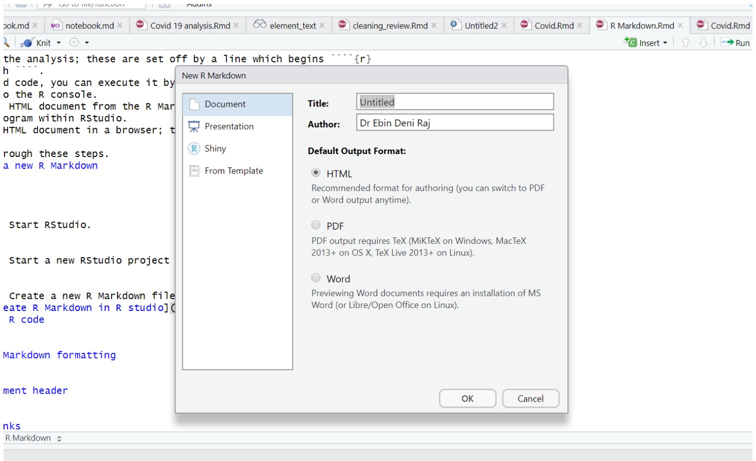
See Figure 3 and Figure 4

The new R Markdown document has some default text, including your name and the date (these can be edited like any text) and is ready for you to enter some text and code; see Figure 5.

Task 5 :This is a sample analysis of gapminder dataset, which we have seen few weeks back. The first step is to load the dataset

See Figure 6. Note that ordinary text is just entered as-is, but R code must be enclosed between two lines:

```
## ```{r}
## code goes here
## ```
```



New R Markdown

Title: Untitled

Author: Dr Ebin Deni Raj

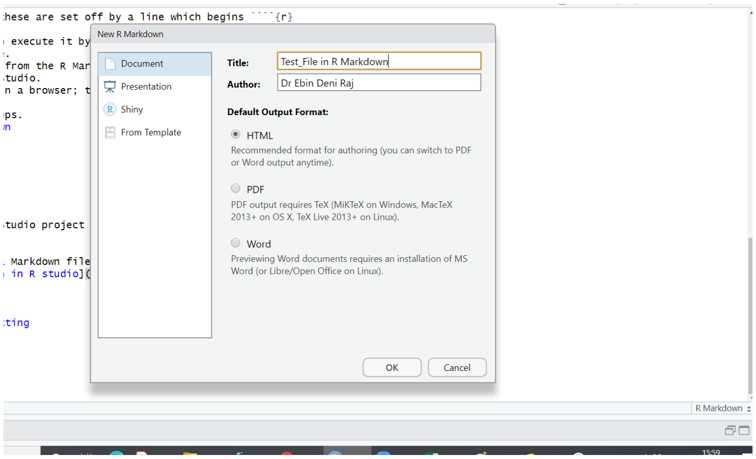
Default Output Format:

● HTML
Recommended format for authoring (you can switch to PDF or Word output anytime).

○ PDF
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

○ Word
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

Figure 2: Options available for creation



New R Markdown

Title: Test_File in R Markdown

Author: Dr Ebin Deni Raj

Default Output Format:

● HTML
Recommended format for authoring (you can switch to PDF or Word output anytime).

○ PDF
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

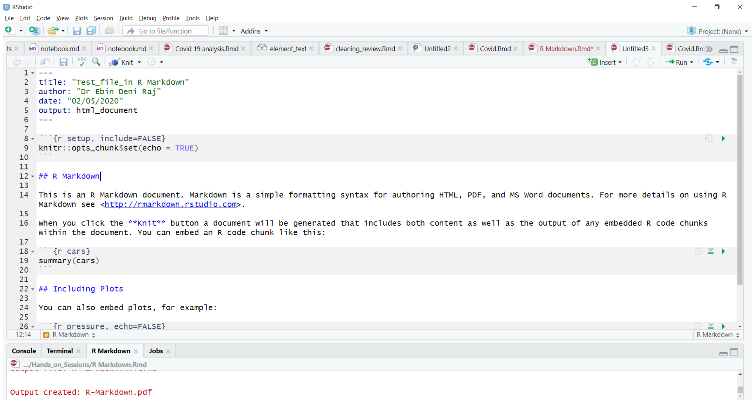
○ Word
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

Figure 3: Naming the file

Task 6 : Save the R Markdown source document with an appropriate name of your choice. Note that the file extension is automatically .Rmd. See Figures 7 and 8.

So we have a source file with some R code. We can run/execute this code to see the results as we work.

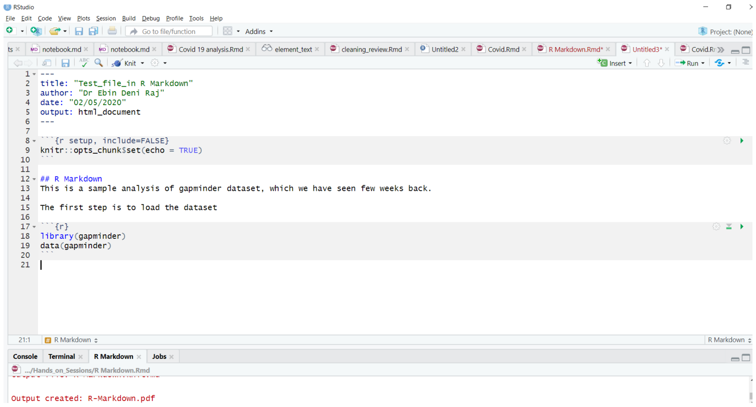
Task 7 : Run the R code chunk from the R Markdown source file and examine the results.



~~~
---
title: "Test_file_in R Markdown"
author: "Dr Ebin Deni Raj"
date: "02/05/2020"
output: html_document
---

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

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R
Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks
within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
~~~

```
Output created: R-Markdown.pdf
```

Figure 4: New document -initial screen



~~~
---
title: "Test_file_in R Markdown"
author: "Dr Ebin Deni Raj"
date: "02/05/2020"
output: html_document
---

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

## R Markdown
This is a sample analysis of gapminder dataset, which we have seen few weeks back.

The first step is to load the dataset

```{r}
library(gapminder)
data(gapminder)
```

~~~

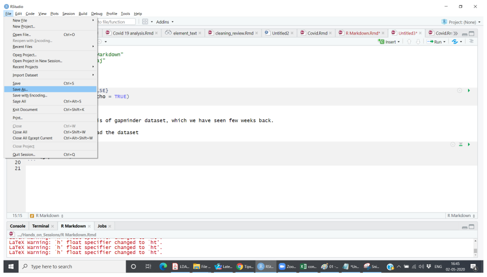
```
Output created: R-Markdown.pdf
```

Figure 5: Entering code and text to the R Markdown file

Compiling R Markdown code

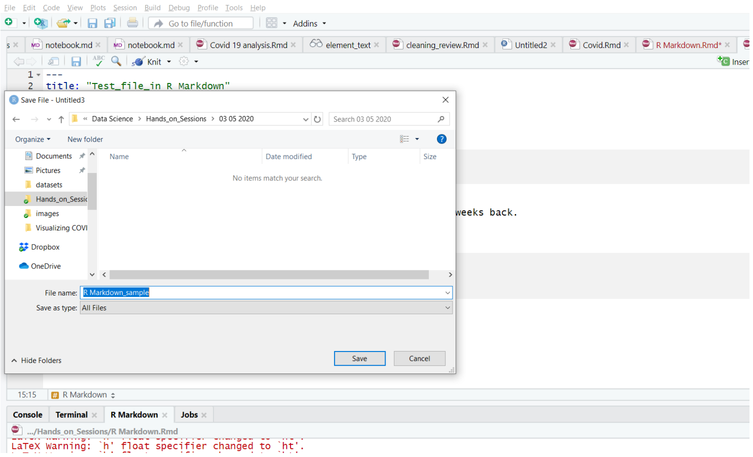
Task 8 Compile the R Markdown source to an HTML document.

The resulting HTML file will open in a new window; . You can also open this in a browser or send it to someone else by e-mail.



```
LaTeX Warning: `h' float specifier changed to `ht'.
LaTeX Warning: `h' float specifier changed to `ht'.
LaTeX Warning: `h' float specifier changed to `ht'.
```

Figure 6: Task 6: Save the R Markdown source document with an appropriate name



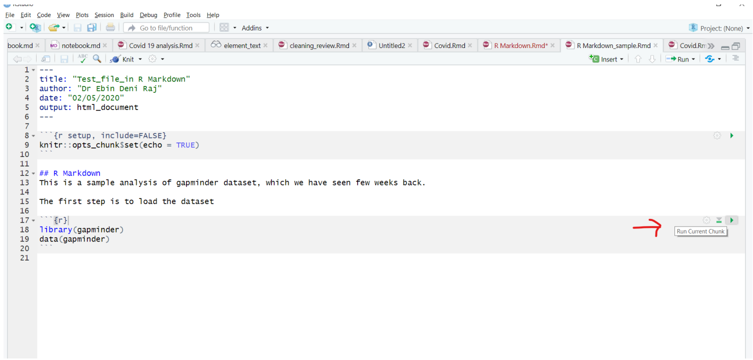
```
---
title: "Test_file_in R Markdown"
```

File name: R Markdown_sample

Save as type: All Files

```
LaTeX Warning: `h' float specifier changed to `ht'.
```

Figure 7: Give appropriate file name



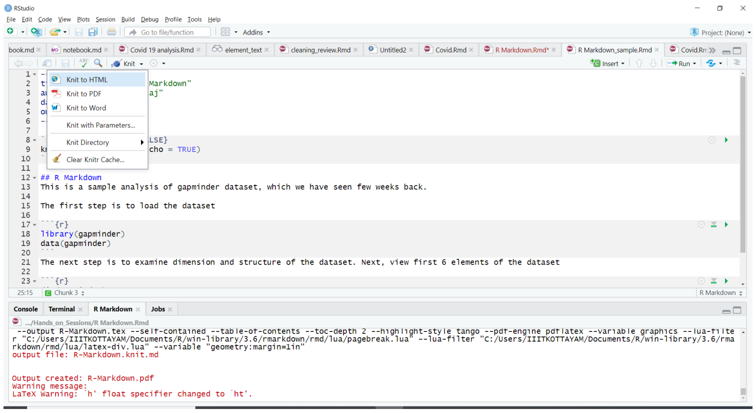
~~~
---
title: "Test_file_in R Markdown"
author: "Dr Ebin Deni Raj"
date: "02/05/2020"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
~~~

R Markdown
This is a sample analysis of gapminder dataset, which we have seen few weeks back.

The first step is to load the dataset

```{r}
library(gapminder)
data(gapminder)
```

```

Figure 8: Run the R code chunk from the R Markdown source file and examine the results
```

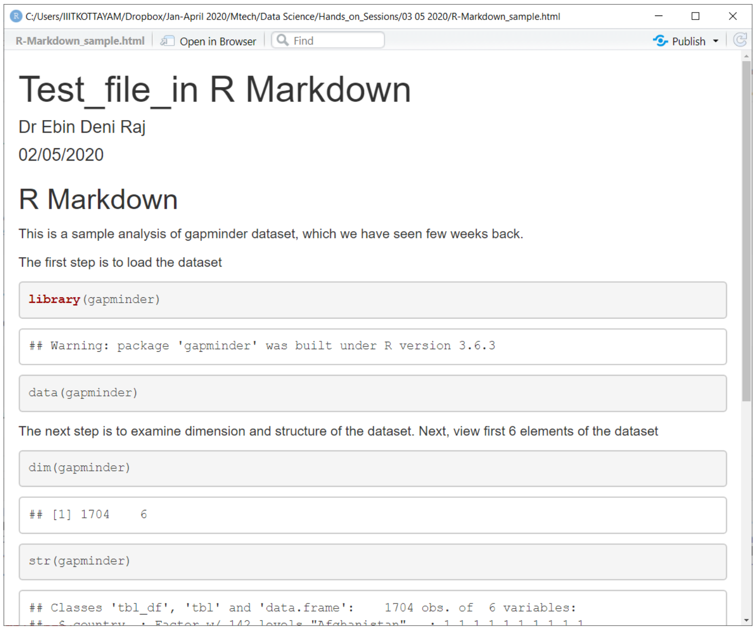


~~~
-
t                                    Markdown"
au                                   aj"
d
ou
-

`                                    LSE}
kn                                   cho = TRUE)
`

## R Markdown
This is a sample analysis of gapminder dataset, which we have seen few weeks back.

The first step is to load the dataset

```{r}
library(gapminder)
data(gapminder)
~~~
The next step is to examine dimension and structure of the dataset. Next, view first 6 elements of the dataset

```{r}
```

```
 --output R-Markdown.tex --self-contained --table-of-contents --toc-depth 2 --highlight-style tango --pdf-engine pdflatex --variable graphics --lua-filte
r "C:/Users/IIITKOTTAYAM/Documents/R/win-library/3.6/rmarkdown/rmd/lua/pagebreak.lua" --lua-filter "C:/Users/IIITKOTTAYAM/Documents/R/win-library/3.6/rma
rkdown/rmd/lua/latex-div.lua" --variable "geometry:margin=1in"
output file: R-Markdown.knit.md


Output created: R-Markdown.pdf
Warning message:
LaTeX Warning: `h' float specifier changed to `ht'.
```

Figure 9: Compile the R Markdown source to an HTML document as shown in the fig. Click on Knit to HTML


C:/Users/IIITKOTTAYAM/Dropbox/Jan-April 2020/Mtech/Data Science/Hands_on_Sessions/03 05 2020/R-Markdown_sample.html
R-Markdown_sample.html
Open in Browser
Find
Publish

Test_file_in R Markdown

Dr Ebin Deni Raj

02/05/2020

R Markdown

This is a sample analysis of gapminder dataset, which we have seen few weeks back.

The first step is to load the dataset

```
library(gapminder)
```

```
## Warning: package 'gapminder' was built under R version 3.6.3
```

```
data(gapminder)
```

The next step is to examine dimension and structure of the dataset. Next, view first 6 elements of the dataset

```
dim(gapminder)
```

```
## [1] 1704    6
```

```
str(gapminder)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1704 obs. of  6 variables:
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1
```

Figure 10: You can view the HTML file as shown here

Word or pdf output

Notice the line output: html_document in the beginning of the Rmd file. It implies that output is written to a html-file. If you have MS Word installed, then it is easy to have the output generated as a docx-file instead: Simply write word_document instead of html_document. Try it!

If you prefer pdf-output, you write pdf_document, but you need to install a version of the TeX program at your computer: MikTex for Windows, MacTeX for mac. If you knit without having TeX installed, then you get an error message which includes links for installation of TeX. Notice that, for MikTex it says that it is important to choose Complete rather than Basic installation, but that appears not to be possible. Fortunately, the problen can be fixed: When you are asked if you want to Install packages on the fly, then answer Yes rather than Ask me first. You must restart RStudio when you have installed TeX (before you can knit to pdf).

Advanced Markdown formatting

- Embed Code Use knitr syntax to embed R code into your report. R will run the code and include the results when you render your report.
- Surround code with back ticks and r. R replaces inline code with its results

Other features will be discussed during the hands On session

Chunk options

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.

YAML document header

The header of an R Markdown file uses the 'YAML' markup language2 to specify what type of document to build from the R Markdown. It is set off by three dashes – before and after. When you create a new R Markdown document a minimal YAML header is created for you:

```
---
title: "My new analysis"
author: "A Nonymous"
date: "1/24/2018"
output: html_document
---
```

The output key: value pair can be used to specify nicer formatting, including document themes, a table of contents, numbered sections, and options for figures. For example:

```
---
title: "My new analysis"
author: "A Nonymous"
date: "1/24/2018"
output:
  html_document:
     number_sections: TRUE
     theme: "spacelab"
     toc: TRUE
     fig_height: 6
     fig_width: 6
---
```

Note the use of white spaces to indent the options. A good introduction and list of options is in the *R Studio on-line documentation*

Some Good practices

- Don't knit all the time. Instead, run code lines and R chunks without knitting while you are figuring out how they should be.

- Do not put all your R commands into one big R chunk. Instead, split it into well-defined smaller chunks, which you may even give names. Investing effort in choosing good chunk names will pay off in terms of structuring your R code.

- The code in Rmd-file must be self-contained in the sense that you cannot use datasets (or other objects) that you have imported "outside" the Rmd-file. You therefore have to include the commands for data import in the file.

- If no output is generated, then read the error message. It is not always easy to read for a beginner, but at least you are informed where the problem occurs.

- It is possible to "cache" R trunks such that the commands are not rerun every time you knit. This is a nice feature if you have time consuming computations.

- As always: Make sure to organize your file in an appropriate way, save the file often, and make sure to save only relevant commands (not all the stuff you played around with at preliminary stages).

- You can open the files in other programs than RStudio, e.g. your favourite browser for html-files or your favourite pdf viewer for pdf-files.

Useful Links

R Markdown home page

RMarkdown Cheat Sheet

RMarkdown Reference Guide