

# CHAPTER 6

## Programmatic Reporting

### OBJECTIVE

Programmatic reporting in the context of data analytics, is the process by which you can seamlessly take raw input data and generate a final report in Word or Pdf in one click. In this chapter, you will learn how to write R scripts that can take your input data and produce a final report in HTML, Word or Pdf. You will also get a glimpse into spreadsheet automation with R. This chapter presents an overview of the techniques for automating various Excel and Google Sheets tasks.

### Contents

6.1	<i>Introduction</i>	225
6.2	<i>Installing R Markdown</i>	226
6.3	<i>Creating Word, PDF and HTML Reports</i>	227
6.3.1	<i>Create a New R Markdown Document</i>	228
6.3.2	<i>Processing R Markdown Documents to Create Reports</i>	231
6.3.3	<i>Adding R Code to R Markdown Documents</i>	234
6.3.4	<i>Review of the R Markdown Template</i>	236
6.3.5	<i>R Markdown Document Example: laborforce.Rmd</i>	238
6.4	<i>Spreadsheet Automation</i>	243
6.4.1	<i>Data Analysis Report in Excel</i>	244
6.4.2	<i>Using Microsoft OneDrive</i>	246
6.4.3	<i>Connecting R to OneDrive with Microsoft365R</i>	251

6.4.4	<i>Data Analysis Report in Google Sheets</i>	263
6.4.5	<i>Connecting R and Excel</i>	265
6.4.6	<i>Connecting R and Google Sheets</i>	267
6.5	<i>Concluding Remarks</i>	269

---

## 6.1 Introduction

---

You can and should consider using R for creating and updating your data analysis reports in Microsoft Word, PDF, HTML, or even in MS Excel and Google Sheets formats. This is particularly relevant if your reports require special formatting and are released on a routine basis. The document formats I have just listed are widely used by nontechnical professionals, which makes the distribution of your analysis convenient. RStudio allows you to go from raw data to a final and polished report with one click. In this chapter, I will show you all the steps from reading your input data, to creating your first report in each of these 5 formats. However, this chapter will focus primarily on the Word, PDF and HTML formats. For Excel, you will find a more detailed treatment of programmatic reporting in chapters 7 and 8. In chapter 9, I will show you how to connect R with Google Sheets to automate your reports.

Producing your data analysis report in Word, PDF, or HTML is recommended if it is meant to be viewed only. However, if additional calculations must be performed on the data tables presented in your report, then releasing it in a spreadsheet format (MS Excel or Google Sheets) is recommended.

R often provides several ways of performing a task. To produce your reports in Word, PDF or HTML directly from R, I recommend using of the R Markdown package named `rmarkdown`. For reports in MS Excel, I recommend using the `xlsx` and `openxlsx` packages. Finally, for Google Sheets, I recommend using the `googlesheets4` package. Of these packages, only `rmarkdown` is discussed in details in this chapter. While the remaining 3 packages `openxlsx`, `xlsx` and `googlesheets4` are reviewed to a limited extent in this chapter, their in-depth treatment is devoted to chapters 7, 8, and 9 respectively.

Section 6.2 shows you how to set up R Markdown. In section 6.3, you will learn how to create your first Word, PDF, and HTML analysis report from within RStudio, after analyzing a dataset. Creating your analysis in a spreadsheet (Excel or Google Sheets) is a topic that I discuss in section 6.4.

## 6.2 Installing R Markdown

---

Markdown is a markup language just like HTML. A typical Markdown document is a combination of plain text with special tags and embedded R code. After processing, the embedded R code is executed and the Markdown document can be rendered in various formats including HTML, PDF or Word. You will not use the Markdown language in this chapter. Instead, you will use “R Markdown,” which is an extension of the Markdown syntax designed to enable R code to be embedded in the text.

Being able to create R Markdown documents has been one my key motivations for using R. The R Markdown document is rich. It can be used to save, and execute code to generate high quality analysis reports that you can share with your audience. You can also mix regular text and mathematical equations in the same document. In this chapter, I will show you how to quickly create an R Markdown document and automate the reporting of your analysis results. You can access a more comprehensive documentation of the R Markdown package online using the following link: <https://rmarkdown.rstudio.com/>.

The very first thing to do before you can create your analysis report, is to install the R Markdown package from the RStudio console as follows:

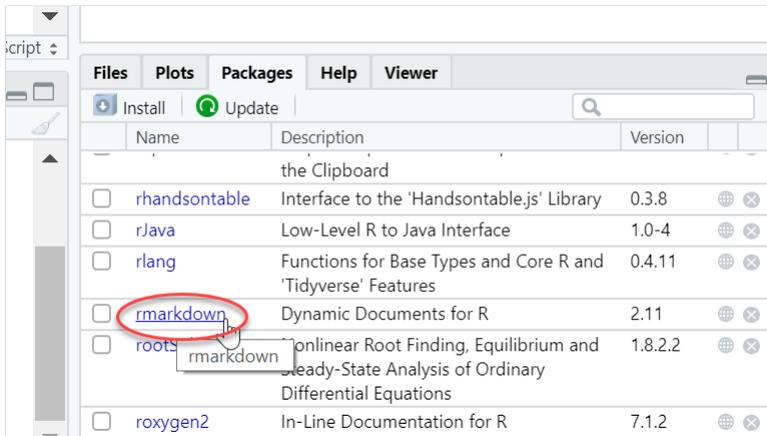
```
> install.packages("rmarkdown")
```

Verify that the "rmarkdown" package installation has been successful. You would do this by checking that `rmarkdown` appears on the list of installed packages. For this, select the "Packages" tab on the lower right pane as shown in Figure 6.1. You also see the package description “Dynamic Documents for R” as well as the package version number 2.11.

In addition to the R Markdown package, you will need the `knitr` package. Install it from the RStudio console as follows:

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

What do you need the `knitr` package for? Well, a typical R Markdown document contains a mixture of text and R code. Therefore, you will need an engine that can process this new compound. And this engine is the `knitr` package. Each of the code chunks found in your R markdown document will be evaluated, and the output (numbers, figures or equations) inserted in the final document if requested.



**Figure 6.1:** Verifying the installation of the `rmarkdown` package

After installing the `rmarkdown` and `knitr` packages, you are now ready to start producing R Markdown documents that you can render in Word, Pdf, or Html.

## 6.3 Creating Word, PDF and HTML Reports

The following link <https://bit.ly/3ye5j5d> contains an analysis report, which shows the estimated and projected size of the US labor force for the years 2000, 2010, 2020, and 2030. The input dataset used to create this report is a `.csv` file named `labor-force-stats.csv`, which you can download using the link <https://bit.ly/3R0y8ui>. In this section, you will learn how to use this input dataset to create the US labor force report, all that from within RStudio. You will create what is known as an R Markdown file, which you will process

to generate a report in various formats.

There are 3 main advantages for producing such a report from within RStudio:

- 1 *Speedy Report Creation:* Once you prepare the initial R Markdown file, you can use new input data to generate the statistical report in a new format with one click. PDF, HTML and Microsoft Word are the 3 formats I consider in this section.
- 2 *Reproducibility:* If you have a new input data file, your reports can be updated with all the formatting in one click. Consequently, you will reap the biggest benefits from using R markdown if your analysis report will need to be updated with new numbers at a later time.
- 3 *Documentation:* The R Markdown document you create lays out all the steps from the input dataset leading to the final report. Your work can be perused by colleagues, and errors can be corrected with a short turnaround time.

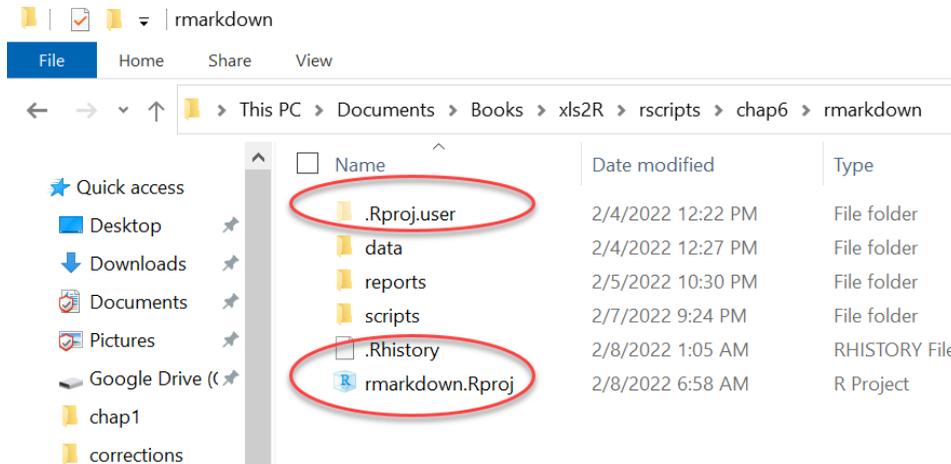
### 6.3.1 Create a New R Markdown Document

You will now create an R Markdown file, save it with the extension `.Rmd`, and process it afterwards to generate the final statistical report. I recommend that you implement the following steps:

- 1 Create a project directory called `rmarkdown`. This directory can be a subdirectory of any other main directory and can be given any name you want. It is a place where all files related to your analysis will be stored.
- 2 Create an RStudio project to link to the newly-created project directory. Go to `File > New Project... > Existing Directory`. Then navigate through directories to select your project directory (in this case `rmarkdown`) before clicking on `Create Project`.

RStudio will then create 2 files (`.Rhistory` and `rmarkdown.Rproj`) and a subdirectory `.Rproj.user` as shown in Figure 6.2.

- 3 I advise that you also create 3 subdirectories `data` (for input data), `reports` (for statistical reports), and `scripts` (for R Markdown files). This recommendation will help you better organize your material.

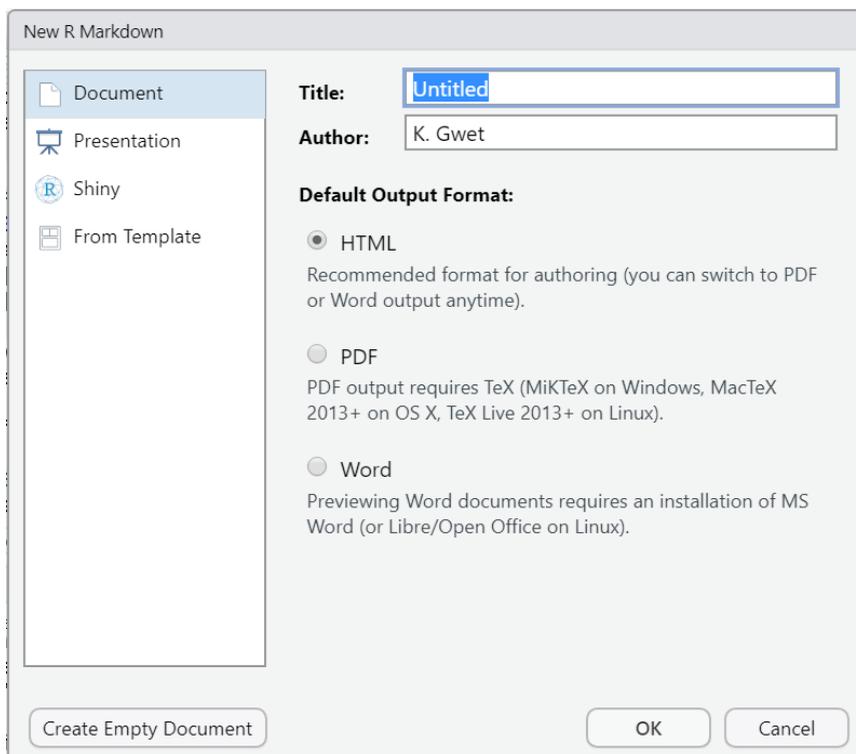


**Figure 6.2:** Project Directory rmarkdown

After getting this prep work out of the way, let us go straight to the creation of our first R markdown document. Go to **File > New File > R Markdown**. A window similar to that of Figure 6.3 will pop up as a result. On this window, you are expected to provide some basic document information that I am now going to review.

- *Document Title:* You may provide a document title. Although this is optional, I decided to key in the title “US Projected Labor Force: 2000-2030.”
- *Author:* The author’s name is also optional. I decided to add my own name “K. Gwet.” Feel free to add yours.
- *Default Output Format:* By default, HTML is selected as the output format, the other alternative formats being PDF, and Word. You can leave

HTML as the default format for now. You can change this output format anytime later.



**Figure 6.3:** Creating a new R Markdown File

On the left vertical bar of Figure 6.3, you will keep the default selection "Document", the other options being out of scope for this book.

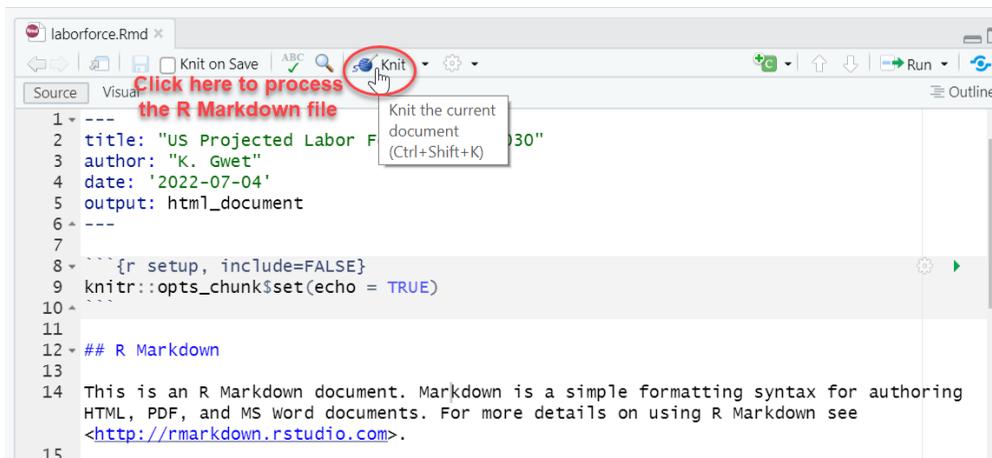
After clicking on the OK button, a new `.Rmd` document named `Untitled1*` will be created. It is a small and fully-functional R Markdown template that gives you a general view on how your own documents should be structured. I will briefly review the most important components of this R Markdown file, before building the `laborforce.Rmd` file in section 6.3.5.

The first thing to do at this stage is to save the current R Markdown document in the `scripts` subdirectory under the name `laborforce.Rmd`.

*The next important thing to do is to close your current R session. Then, open the project file `rmarkdown.Rproj`. From now on, you will be working inside an RStudio project. Script files will know where to get input data files and where to write analysis reports.*

### 6.3.2 Processing R Markdown Documents to Create Reports

In this section, I will use the `laborforce.Rmd` file created in the previous section to produce the final report in HTML format. Open the `.Rmd` file `laborforce.Rmd`, in case it is not already opened. Figure 6.4 shows a portion of this file in the RStudio Source pane. Note the presence of the red-circled Knit button<sup>1</sup> on RStudio's Integrated Development Environment (IDE). This is the button you should click on to process the R Markdown file. As soon as you click on the Knit button, RStudio will render and preview an HTML output using its integrated previewer as seen in Figure 6.5.



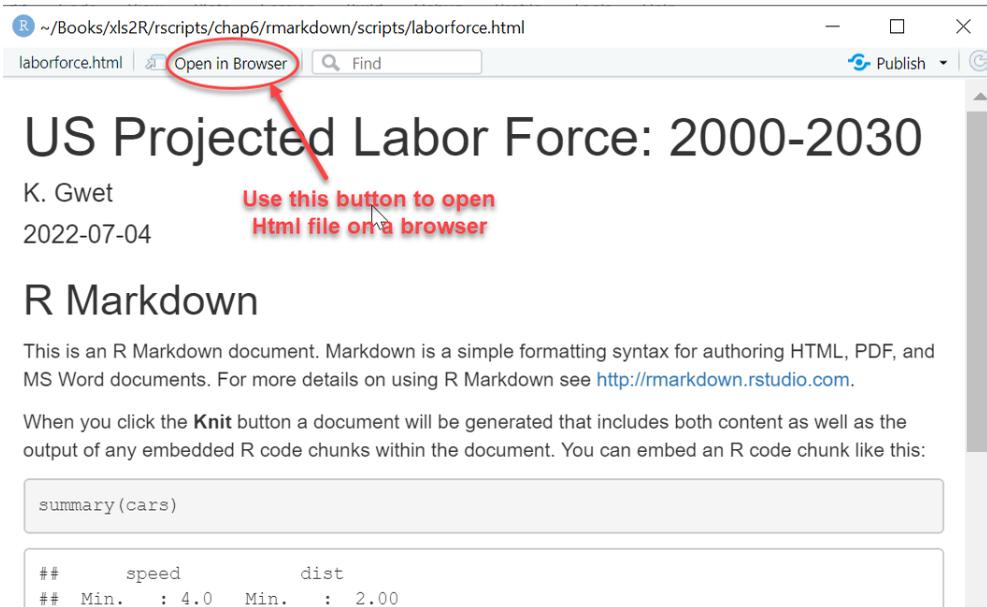
**Figure 6.4:** Temporary Labor Force R Markdown file

To save this output, you need to open this HTML file on a browser by clicking on the "Open in Browser" button at the top left side of RStudio's

<sup>1</sup>If you do not see the Knit button, make sure the `knitr` package has been installed as recommended earlier.

viewer shown in Figure 6.5. You can view this entire HTML file using the following link <https://bit.ly/3bS88Bv>.

Remember that HTML is the default format that you chose earlier (c.f. Figure 6.3). A direct consequence of this choice is that RStudio inserted line #5 in the R Markdown file (c.f. Figure 6.4), defining the default output format with the command `output: html_document`.

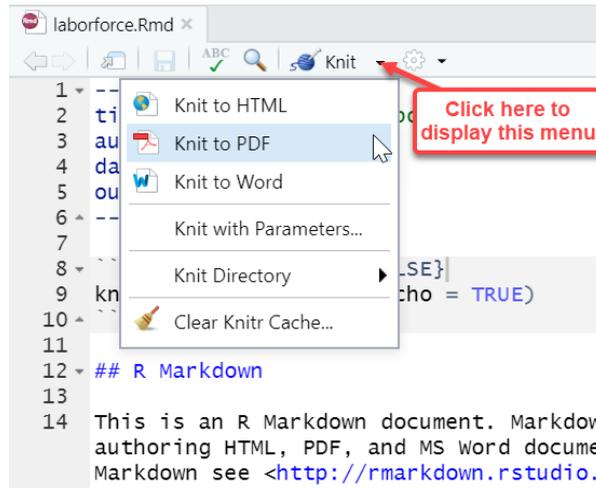


**Figure 6.5:** Laborforce.html displayed in RStudio Preview Window

I now want to review the most important components of the `laborforce.Rmd` R Markdown file.

- The first 5 lines of this R Markdown file are automatically generated by RStudio based on the information you provided on the dialog form of Figure 6.3. These are the title, author's name, the date, and the default output format.
- Suppose that you want to generate a PDF output file instead of an HTML file. Click on the tiny down arrow on the right side of the `Knit` button

to display the dropdown menu as shown in Figure 6.6.



**Figure 6.6:** Selecting additional output formats for the R Markdown file

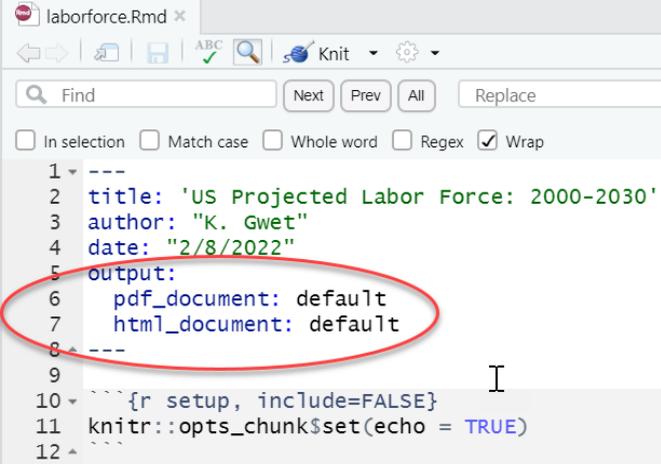
Then select the **Knit to PDF** menu item. This action will process the same R Markdown file before previewing the PDF file. You can now save this PDF file by displaying the PDF viewer menu (click on the 3 bars on the top left side of the PDF viewer), and going to **File > Save As...**

After creating the PDF file, the R Markdown file is updated by RStudio as shown in Figure 6.7. Note that what was line #5 in Figure 6.4 is now replaced with lines #5, #6, and #7. If you decide later to "Knit to Word", then RStudio will insert the command `word_document: default` as the new line #6. Note that each time you click on the **Knit** button, RStudio will render the text according to the content of line #6.

The portion of the R Markdown document, which is surrounded by delimiter marks `---` is referred to as the **YAML**<sup>2</sup> header.

<sup>2</sup>The YAML header contains metadata that tells R Markdown how to display the document. What that acronym YAML stands for is unclear, although "Yet Another Markup Language" has been suggested as a possibility.

If you are going to manually modify the YAML header, you need to always remember that indentation is mandatory for the commands `pdf_document`, `html_document`, or `word_document`. Without it, the rendering of your document may not work.



```

1 ---
2 title: 'US Projected Labor Force: 2000-2030'
3 author: "K. Gwet"
4 date: "2/8/2022"
5 output:
6   pdf_document: default
7   html_document: default
8 ---
9
10 {r setup, include=FALSE}
11 knitr::opts_chunk$set(echo = TRUE)
12

```

**Figure 6.7:** Modified R Markdown file `laborforce.Rmd` after rendering in the PDF format

Other than the YAML header, an R Markdown document is generally a mixture of text and R code. When you process your `.Rmd` file, R Markdown (more precisely `knitr`) will run each R code before embedding the results in the final report in Word, PDF or HTML. You may decide to include or exclude the code from your final report<sup>3</sup>. In the next section, I discuss how you would include R code into your R Markdown document.

### 6.3.3 Adding R Code to R Markdown Documents

There are 2 ways in which you can include R code in an R Markdown document. One way is to use a *chunk of code* aimed at producing results in a separate paragraph. Another way is to use an *inline R code*, which is embedded

<sup>3</sup>Reports that are intended for a large audience will typically exclude all R code. But a draft technical report to be perused by colleagues may include code.