

CHAPTER 1

Introduction

OBJECTIVE

This chapter is introductory and discusses the main benefits of using R to leverage Excel analytical capability. It gives you a high-level overview of the advantages and disadvantages for using both R and Excel, and explains how connecting both can dramatically increase your productivity as an analyst.

Contents

1.1	<i>Short Overview of R and Spreasheets</i>	8
1.2	<i>Spreadsheets: Advantages and Disadvantages</i>	9
1.2.1	<i>Advantages of Excel Spreadsheets</i>	10
1.2.2	<i>Problems with Spreadsheets</i>	13
1.3	<i>Using R to Automate Spreadsheets</i>	19
1.3.1	<i>Interaction Between R and Excel</i>	21
1.3.2	<i>Creating and Sharing Analytic Reports Programmatically</i>	24
1.4	<i>Concluding Remarks</i>	27

1.1 Short Overview of R and Spreadsheets

I love spreadsheets (Excel and Google Sheets) nearly as much as I love the R software. Using both has always increased my productivity. R and spreadsheets are complementary in my views. Therefore, I will not advise you to choose one and ignore the other. Spreadsheets are about simplicity, whereas R is about flexibility. Excel spreadsheets for example, are intuitive and their basic use requires limited training. R on the other hand, is so versatile that the types of tasks it can undertake and the different ways these tasks can be undertaken are astonishing.

Many experienced Excel users often resist the idea of getting into R due to their perceived difficulty of coding. This perception is deceptive and this resistance is unjustified. In the beginning, Excel was a simple computerized version of an accountant worksheet before it became the highly sophisticated software we know today. It even offers a built-in fully-fledged programming language known as Visual Basic for Applications (VBA). R on the other hand, followed a different development path. It started as a pure programming language for programmers, before becoming largely run today through a very popular and user-friendly “Integrated Development Environment (IDE)” known as RStudio. RStudio does not require advanced programming skills. I do not see a single reason why one would want to use R today without the RStudio interface. RStudio makes the use of R much simpler. Because of these different development paths, R has always been associated with computer programming and not Excel. Both however, provide very powerful programming languages for advanced users.

I am an experienced Excel VBA and R programmer and have developed advanced software applications with both languages. I like to give you a glimpse into how far you might take your R and Excel skills, if you are ambitious. The link <https://agreestat.com/examples/> shows you an example of an Excel program I wrote a few years ago. I developed this software in its entirety

based on a single Excel workbook and its built-in tools. I developed a similar software in the cloud using only R and some of its many packages. To see how it works, you may check out the link <https://agreestat.com/examples360/>. The development of such applications is for advanced users only, and out of scope for this book. In this paragraph, I wanted you to see what you can achieve with both Excel and R. The ultimate goal of this book is more modest. It is to show how you can use R to leverage Excel data management and analysis capabilities in a very significant way.

For simplicity, you can see R as being to RStudio what VBA is to Excel. This comparison is somehow simplistic¹, but would give you a good idea of the nature of the R-RStudio relationship. For those coming from the Excel world, it is recommended to use R only with the RStudio interface. To summarize, I recommend to always use RStudio with the understanding that R needs to be installed before RStudio can work.

In the next few sections, I will discuss some key advantages and disadvantages for using Excel and R. I will then reveal why I will always use both Excel and R to increase productivity by combining the simplicity of spreadsheets, and the flexibility and power of R.

1.2 Spreadsheets: Advantages and Disadvantages

Spreadsheets are popular, very popular, have been enjoying that popularity for a while and are expected to continue enjoying it in a foreseeable future. Because of this factor alone, you can only ignore spreadsheets at your own risks. However, this popularity should not make you oblivious to Excel's limitations. In this section, I will explain why I like spreadsheets and also why I cannot rely solely on them for my data analysis works. This will allow you to understand the strengths fueling Excel popularity and the limitations that motivated me and others to consider alternatives tools.

¹As a matter of fact Excel does not depend on VBA as much as RStudio depends on R. Excel used to exist prior to VBA being added to it, whereas RStudio was built to run R.

1.2.1 Advantages of Excel Spreadsheets

Excel is Intuitive

The main reason I like spreadsheets is that they are intuitive and allow anybody to conduct basic exploratory data analysis with no or limited training. Look at Figure 1.1, which depicts a small portion of an Excel worksheet where the number 14 was typed in a cell. Actually 14 is located in cell C2 of a worksheet named “Sheet1.” If you wanted to do the same thing in R you will need to execute the command `Sheet1[2,3] <- 14`, which means that you want to assign number 14 to a location in the data file “Sheet1,” defined by row 2 and column 3.

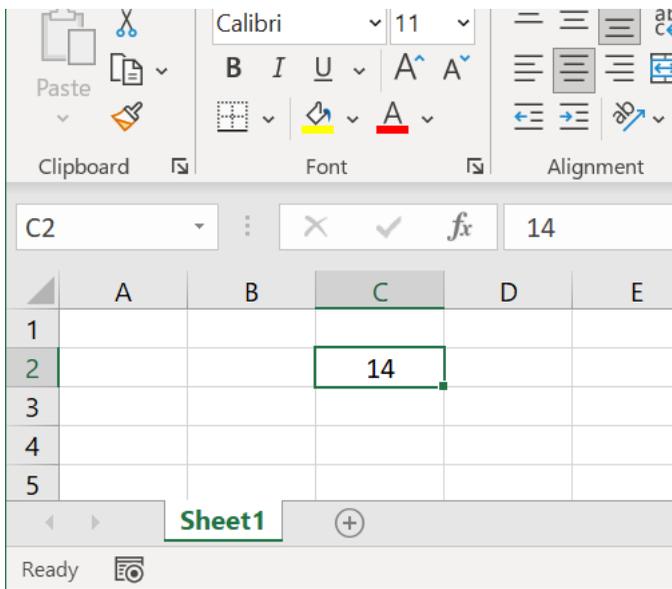


Figure 1.1: Cells in an Excel Worksheet named “Sheet1”

In Excel, you can just type your number in any random cell and build a simple dataset without giving it a thought. You can see here one key reason I do not stay away from spreadsheets. I can easily create a small dataset from scratch in Excel by grabbing a few data points from different places, copying

and pasting. I would then import it from within R for further analysis, rather than going through the challenges of creating it directly in R.

Sometimes, I start my data analysis task with Excel. I will create, or visualize my dataset (at least a portion of it) in Excel to have a first glance at its content. I could perform basic calculations to further explore the dataset, before saving it in a CSV (Comma delimited) text format. I will then read the CSV file from R and analyze it before exporting the results to Excel, Word or PDF format for distribution. This is the workflow that has worked best for me, and one that I highly recommend.

What You See Is What You Get (WYSIWYG)

In Excel, you do not need to conceptualize the task at hand. No need to think in terms of variables and the type of data that will be assigned to them (numbers, letters, ...). No need to deal with any level of abstraction, because of the What You See Is What You Get (WYSIWYG) layout. The numbers represent what you see and are located where you see them. If you want to see them elsewhere, you can move them there. The only time casual Excel users need to worry about where their numbers are located is when using formulas. It is then necessary to type things such as “=D3 + E3,” where D3 and E3 are the coordinates on the data grid associated with the 2 data points you want to add.

Another appealing aspect of spreadsheets is their reactive computation model for worksheets that are already set up with formulas. As soon as you modify some of your input data, the entire worksheet is automatically updated. This feature is very convenient for conducting a quick ad hoc analysis.

Widespread Availability

Excel is likely the only data analysis tool most professionals will ever use. It does not matter which tool you used to do your analysis, if you are going to

share preliminary results with your colleagues, you will need to use a spreadsheet (Excel or Google Sheets). A final report may be prepared in HTML, PDF, Word or any other Office product such as PowerPoint.

When conducting a data analysis task, you can make significant gain in productivity by transferring input data from Excel to R, analyzing it in R before exporting the results back to Excel. But, you need to make this process as reliable as possible. In this approach, Excel is used as a convenient of sharing smaller data tables with others, whereas R allows you to develop more sophisticated data pipelines in addition to using superior data analysis tools.

Here are 2 of the most valuable Excel resources I have ever seen:

- If you are primarily a non-programmer Excel user, then the website <https://www.xelplus.com/> is likely one of the best resources you can find on the Internet. Although I have been using Excel for a long time, I still learn new techniques from this very instructive website. You will also have the opportunity to watch a large number of inspiring videos.
- Excel VBA programmers will hardly find a better resource than the following website: <https://www.excelforfreelancers.com/>. Without this resource, it would have taken me a lot longer to reach my current proficiency level in VBA programming. If you are already an Excel VBA programmer, then you can greatly benefit from this resource. However, if you want to learn computer programming (even with Excel), then I would not recommend VBA. It is unlikely that it will ever be used on the web. Spreadsheets are used in the cloud more often now than ever before. A better option would be to start learning the newly-released LAMBDA function of Excel. Alternatively, you can just learn to program in R.

1.2.2 *Problems with Spreadsheets*

As much as I love MS Excel, it has a few problems inherent to spreadsheets in general that can hinder productivity in a professional setting. In this section, I am going to review some of these problems.

Documenting your Work

Your colleagues at work may find it difficult to peruse your spreadsheet, especially if it contains complex calculations. It is because all the changes you make to an Excel worksheet do not leave a trail behind. You may not even know yourself what changes you previously made to your own worksheet that led to the final analysis. Although worksheet cells can be commented, comments embedded in a cell are of limited use. Here are a few problems with cell documentation:

- Excel formulas are difficult to read. Documenting each cell with a formula can be tedious if there are many of them. Moreover, cell comments are expected to be short. Otherwise, their readability becomes problematic.
- Cell comments are hidden by default, although you may choose to display them. Either option is bad if the number of commented cells is large. Hidden comments will require the reviewer of your spreadsheet to hover every cell to read them. Displayed comments on the other hand, will make the worksheet look messy.

To summarize, collaboration based on spreadsheets can be challenging. If there is a need to coordinate activities with others, you may need to consider alternative tools.

Each time you see a number in a spreadsheet cell, you have no way of knowing how many times it was changed, by who and for what purpose. This problem becomes particularly worrisome if the spreadsheet must change hands many times. Note that there is always a way to lock portions of a worksheet.

This cannot be a desirable option as others will not have the opportunity to add their comments.

With the R software, documenting your work is simple. This simplicity is due primarily to the exclusive use of text files in R. That is, you can freely add unlimited comments anywhere your R commands are used. In chapter 10, you will learn the important advanced and modern concept of version control, which allows you to keep track of the different versions of your work in an efficient way.

Managing Large Repeatable Projects

When your worksheet contains thousands of rows of data, the use of spreadsheets becomes less appealing. In fact, nobody can realistically look at a 10000 by 600 data table with the naked eyes and be able to learn something of value out of this activity. Therefore, a moderately large dataset makes the spreadsheet less appealing. You no longer have the ability to see everything. Even entering formulas into cells is no longer trivial. That is, describing data ranges requires that you manually select large portions of the worksheet with the computer mouse. The process becomes error-prone as some cells can be overlooked with no way of knowing it and with potentially serious consequences. A compelling example that illustrates this issue comes from the Reinhart & Rogoff (2 Harvard economists) article in the Times (see [Ryssdal, 2013](#)) on the impact that national debt has on economic growth. You can learn more about this issue from [Fernholz \(2013\)](#).

In R, you will learn to use basic commands, the validity of which can be verified by yourself as well as by others. The different steps of the analysis will be clearly laid out and properly documented. The entire process will be transparent, and everyone can see the details of what was done to produce the results. These results can be updated with the availability of new data.

Even if a worksheet does not contain many rows of data, it could be part of a workbook containing several worksheets. Your input data may even be

spread across several independent Excel workbooks, and other external non-Excel data sources. Using Excel to manage such a complex data pipeline based on functions such as `VLOOKUP`, will be tedious, labor-intensive and error-prone.

The more recent versions of Excel include some sophisticated data analysis tools in the form of add-ins. Two of these tools of interest are the **Power Pivot** and the **Power Query**. Unlike the traditional Excel **Pivot Table**, the **Power Pivot** can analyze data across multiple Excel tables after creating a data model. A data model is a collection of tables and the relationships defining how they can be linked together to form a relational database. **Power Query** on the other hand, will help you gather data from multiple sources, clean and transform it, then prepare it for analysis. It also offers the possibility to automate the whole process by writing a query. For analysts wanting to complete their entire analysis within Excel, **Power Pivot** and **Power Query** will dramatically reduce the time to project completion. However, these tools are still typical Excel solutions, which carry the same weaknesses in the areas of documentation or collaboration.

Many statistical agencies of the US government distribute data in the form of Excel workbooks. For example the Bureau of Labor Statistics (BLS) provides several such tables². It is common for analysts to download several such Excel workbooks to prepare a report, and often need to repeat it. Taking the time to learn how such a task can be completed using R will yield considerable gain in efficiency. In chapter 6, you will learn how to use R to easily generate an analysis report in Word by simply changing the input dataset.

Developing a Good Work Ethic

When you work with Excel, especially with a small dataset, you can always get your analysis done even if your worksheet is not properly structured for analysis. Consider the spreadsheet of Figure 1.2, showing the size of the US labor force in the years 2000, 2010, 2020 and 2030 by gender and age. This

²The BLS website <https://www.bls.gov/emp/tables.htm> provides such Excel datasets

worksheet looks more like a final report than an input dataset to be analyzed. But if you want to summarize this dataset by gender only, such as in Figure 1.3 and by age only such as in Figure 1.4, you can use the SUM function of Excel and manually select the relevant cells with mouse clicks.

	A	B	C	D	E
1	Group	Labor force, 2000	Labor force, 2010	Labor force, 2020	Labor force, 2030
2	Men, 16+ years				
3	16 to 19	4,270	2,992	2,859	2,272
4	20 to 24	7,521	7,864	7,417	6,975
5	25 to 34	17,844	18,351	19,509	18,976
6	35 to 44	20,093	18,119	18,247	20,882
7	45 to 54	16,269	18,855	16,920	17,539
8	55 to 64	7,796	12,103	14,382	13,518
9	65 to 74	2,018	2,971	4,755	6,540
10	75 and older	471	729	1,116	2,092
11					
12	Women, 16+ years				
13	16 to 19	4,001	2,915	2,861	2,373
14	20 to 24	6,730	7,164	7,070	7,078
15	25 to 34	14,911	15,262	16,998	17,205
16	35 to 44	17,473	15,247	15,737	17,590
17	45 to 54	14,802	17,105	15,220	16,162
18	55 to 64	6,561	11,194	12,911	12,922
19	65 to 74	1,487	2,453	3,871	5,700
20	75 and older	336	564	871	1,813

Figure 1.2: Predicted labor force in 2000, 2010, 2020 and 2030 by gender of age range

With small datasets, you can see everything. Your unstructured data and your analysis will be located in the same worksheet. This may give Excel users the false impression that even larger datasets can safely be handled without a better work ethic. In reality, without a more rigorous approach to data analysis, your worksheet can quickly start looking like the messy playground of a troubled child, with numbers all over the place and no way of knowing what

constitute input data, and what constitute the results of an analysis.

B	C	D	E	F
gender	LF2000	LF2010	LF2020	LF2030
All	142583	153888	160744	169637
MEN	76282	81984	85205	88794
WOMEN	66301	71904	75539	80843

Figure 1.3: Predicted labor force in 2000, 2010, 2020 and 2030 by gender

B	C	D	E	F
age	LF2000	LF2010	LF2020	LF2030
All	142583	153888	160744	169637
16 to 19	8271	5907	5720	4645
20 to 24	14251	15028	14487	14053
25 to 34	32755	33613	36507	36181
35 to 44	37566	33366	33984	38472
45 to 54	31071	35960	32140	33701
55 to 64	14357	23297	27293	26440
65 to 74	3505	5424	8626	12240
5 and olde	807	1293	1987	3905

Figure 1.4: Predicted labor force in 2000, 2010, 2020 and 2030 by age range

More powerful tools such as R and larger datasets require a better organization and more discipline. You need to have a more systematic and logical way of doing things. The focus is now on writing precise instructions that the computer must execute. Consequently, you must know where the data is located in the worksheet without looking. This requirement will typically force you to think more about the task at hand before the analysis can begin.

A dataset such as that of Figure 1.2 cannot be analyzed with R. What is the problem then? Here are some of the key problems with that dataset:

- Row #11 for example contains no data. Moreover, rows #2 and #12 only contain data in column A. Unless, there is meaning to such empty rows,

there is no reason why they should there. Again, you need a dataset that a computer can easily interact with. Not one that will please the naked eye.

- A far bigger problem lies in column A, labeled as “Group.” Note that this column contains 2 types of data. These are the gender and age of the labor force participant. When preparing a dataset for “serious” analysis, *the same column should never ever contain more than one type of information*. The rationale being that if you read a random row (or record) of that dataset, you do not want to have to figure out whether you have read the age or whether it is the gender. With such a confusion possible, data analysis will be very difficult.

If you want to analyze Figure 1.2 data in R, you will need to reorganize it as shown in Table 1.2 . As you can see, this table gives each column a clear-cut and unique identity. The label of each column can now be referred to as a *Variable*. Table 1.2 can become a formal R dataset with a record layout defined by Table 1.1 .

Table 1.1 : Record layout of the Labor Force (LF) dataset of Table 1.2

variable #	Maximum Length	variable name	variable description
1	5	Gender	Gender of the LF participant
2	8	AGE	Age of the LF participant
3	6	LF2000	Size of the 2000 LF
4	6	LF2010	Size of the 2010 LF
5	6	LF2020	Size of the 2020 LF
6	6	LF2030	Size of the 2030 LF

Until you can describe your data structure as shown in Table 1.1 , you will not be able to exploit it in R, nor in any other advanced data analysis software.

You will see in subsequent chapters, how to read Table 1.2 data into R. Then, with a few lines of R code, you will be able to create an Excel workbook with 2 sheets, one containing Figure 1.3 data and the other containing Figure

1.4 data. The process of creating these pivot tables in R will be simple, quicker than in Excel with or without Power Pivot.

Table 1.2 : US Labor Force Projections^a

gender	age	LF2000	LF2010	LF2020	LF2030
MEN	16 to 19	4,270	2,992	2,859	2,272
MEN	20 to 24	7,521	7,864	7,417	6,975
MEN	25 to 34	17,844	18,351	19,509	18,976
MEN	35 to 44	20,093	18,119	18,247	20,882
MEN	45 to 54	16,269	18,855	16,920	17,539
MEN	55 to 64	7,796	12,103	14,382	13,518
MEN	65 to 74	2,018	2,971	4,755	6,540
MEN	75 +	471	729	1,116	2,092
WOMEN	16 to 19	4,001	2,915	2,861	2,373
WOMEN	20 to 24	6,730	7,164	7,070	7,078
WOMEN	25 to 34	14,911	15,262	16,998	17,205
WOMEN	35 to 44	17,473	15,247	15,737	17,590
WOMEN	45 to 54	14,802	17,105	15,220	16,162
WOMEN	55 to 64	6,561	11,194	12,911	12,922
WOMEN	65 to 74	1,487	2,453	3,871	5,700
WOMEN	75 +	336	564	871	1,813

^aSource:<https://www.bls.gov/emp/tables/civilian-labor-force-summary.htm>

1.3 Using R to Automate Spreadsheets

In this section, you will get a glimpse into R and some of its key advantages compared to Excel. Chapter 2 will show you step by step how to set up the R computing environment and start undertaking useful tasks. Some of the concepts introduced in this section, may be new to you. They will only be defined in general terms here. But a more detailed discussion will follow in subsequent chapters.

You do not need to be a computer programmer to use RStudio, in the

same way you do not need to be a VBA programmer to use Excel. Although you can develop highly sophisticated applications with RStudio, this book will focus on helping Excel analysts use the power of R to boost their productivity. In my views, doing “serious” R programming amounts to writing R functions that are reused in different places of a larger program. I will not get into function development in this book. If you already know R and want to learn R programming, I suggest [Grolemund \(2014\)](#).

What you will often do as a beginner, is create simple R script files that resolve your problem. A typical script file will contain basic commands and functions available in various R packages. These R packages are modules, developed by third parties to improve R’s analytical capabilities. A small R script file can have only one command such as `x<-12`, which asks R to create an object called `x` and to assign number 12 to it for later use. This is similar to typing 12 into an Excel cell such as B3. A script is a text file containing a set of commands and possibly some comments. The script can be saved, documented and used later to re-execute the saved commands. It can also be edited to incorporate new features.

The R language comes with several built-in functions, the most important of which can be learned fast. Additionally, with R being a free and open-source language³, several independent developers have developed what is known in the R world as packages, some of which expanded the functionality of R in a very impressive way. You may even use some of these packages to develop web applications. But this will require some specialized skills not covered in this book. Throughout this book, you will discover several useful R packages. You will learn how to use them in your analysis tasks.

In this book, a particular emphasis will be put on the interaction between R and Excel, and on automatic report creation in PDF, Word and HTML formats. In section [1.3.1](#), you will see how you can go from Excel to R and back to Excel.

³A software is open-source if the user is granted a license to modify and freely distribute it.

In section 1.3.2, I will present a high-level overview of the process of reading an Excel dataset, analyzing it in R and creating a report in Word. In subsequent chapters, I will also present a more detailed account of the process of creating analysis reports automatically with an R script.

1.3.1 *Interaction Between R and Excel*

Script file 1.1 reads Table 1.2 data⁴, and creates a workbook named “labor force data analysis.xlsx.” This workbook has 2 worksheets named `byage` (see Figure 1.6) and `bysex` (see Figure 1.5). The `byage` worksheet contains Figure 1.4 data, whereas worksheet `bysex` contains Figure 1.3 data. Unless you already have some experience with R, the commands in Script 1.1 will not mean much to you. However, all of them will be reviewed in details in subsequent chapters and you will find them easy to master.

I do not use Excel’s point-and-click approach and its menu-driven interface to manually create the workbook “labor force data analysis.xlsx.” Instead, I use the R language to issue instructions to the computer on how to do this work for me. If I write my script well enough, I will never have to do the exact same work again. Instead, I will execute my script again with new data. Without R or a similar language, you would need to repeat an error-prone manual work in order to recreate this Excel file. As an analyst, it is essential to use your judgment to determine which part of your work must be done in R and which one would be done in Excel. A guiding principle is to consider R in the following situations:

- Excel requires substantial copying and pasting. This activity can become tedious, tiring and possibly error-prone, if carried out over an extended period of time.
- You need to type complex formulas in Excel. Excel formulas are known

⁴This data is stored in the file `labor-force-stats.csv` that you may download using the link <https://bit.ly/3doWEpA>

to be unpleasant to read, as they do not follow the conventional mathematical notations. They are therefore difficult to debug.

- You need to go through a complex menu systems to select the options required to set up your analysis task. Reproducing such an analysis each time the input data changes will be time-consuming.

Note that Script 1.1 will be needed in subsequent chapters, and can be downloaded using the link <https://bit.ly/3pg4LYq>. The Excel file “labor force data analysis.xlsx” can also be downloaded with the link <https://bit.ly/3QMj6r1>

Script 1.1. Calculating column sums of Table 1.2 by Gender, then by Age

```

01 pacman::p_load(readr,dplyr,xlsx)
02 fra.lforce <- read_csv("xls2R/data/labor-force-stats.csv",
03   show_col_types = FALSE)
04 #Summarize LFS data by gender
05 by.gender <- fra.lforce%>%
06   group_by(gender)%>%
07   summarise(across(where(is.numeric),sum))
08 by.gender.tot <- (fra.lforce %>%
09   summarise(across(where(is.numeric),sum)) %>%
10   mutate(gender="All"))[,c(5,1:4)]
11 by.gender <- rbind(by.gender.tot,by.gender)
12
13 #Summarize LFS by age
14 by.age <-fra.lforce%>%
15   group_by(age)%>%
16   summarise(across(where(is.numeric),sum))
17 by.age.tot <- (fra.lforce %>%
18   summarise(across(where(is.numeric),sum)) %>%
19   mutate(age="All"))[,c(5,1:4)]
20 by.age <- rbind(by.age.tot,by.age)
21
22 #-- Creating workbook containing statistics by gender & age
23 wb <- createWorkbook()
24 sht.byage <- createSheet(wb, sheetName="byage")
25 sht.bysex <- createSheet(wb, sheetName="bysex")

```

1.3. Using R to Automate Spreadsheets

```

26 addDataFrame(as.data.frame(by.gender),sht.bysex,row.names=FALSE)
27 addDataFrame(as.data.frame(by.age),sht.byage,row.names=FALSE)
28 saveWorkbook(wb, "xls2R/data/labor force data analysis.xlsx")
    
```

End of Script

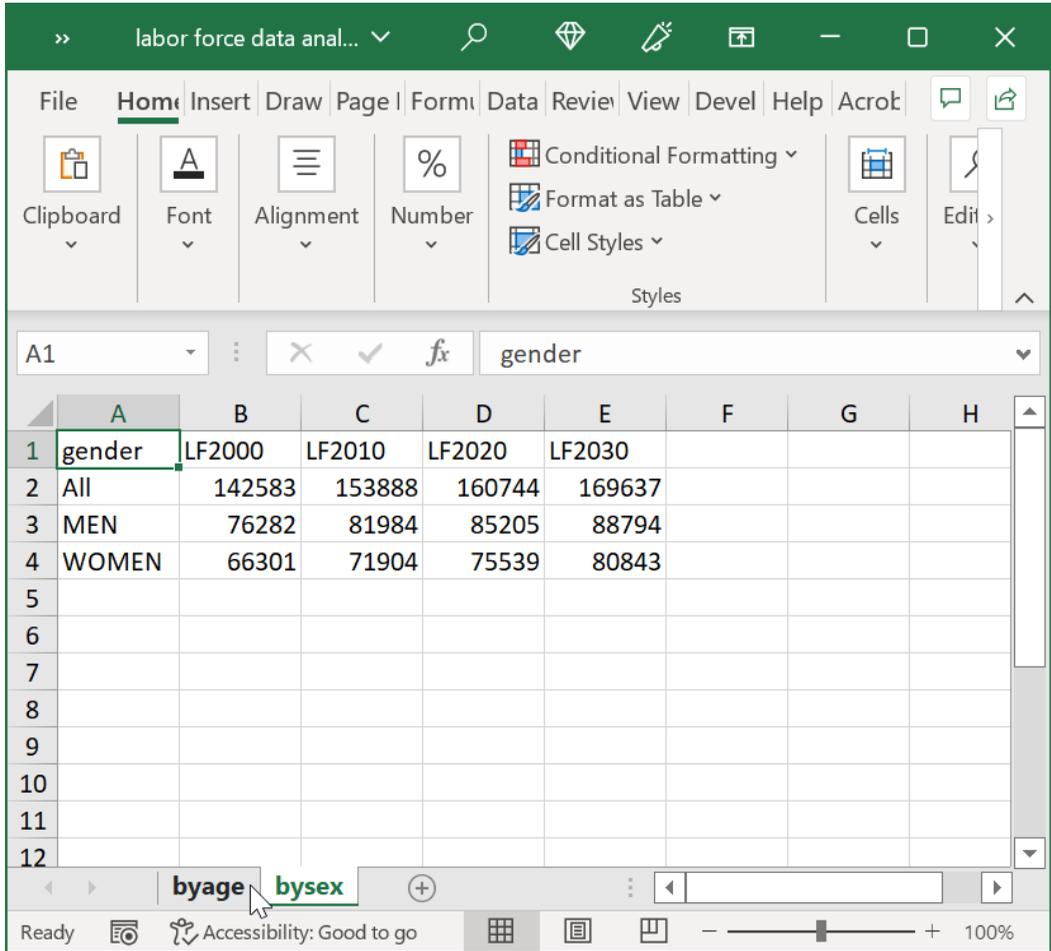
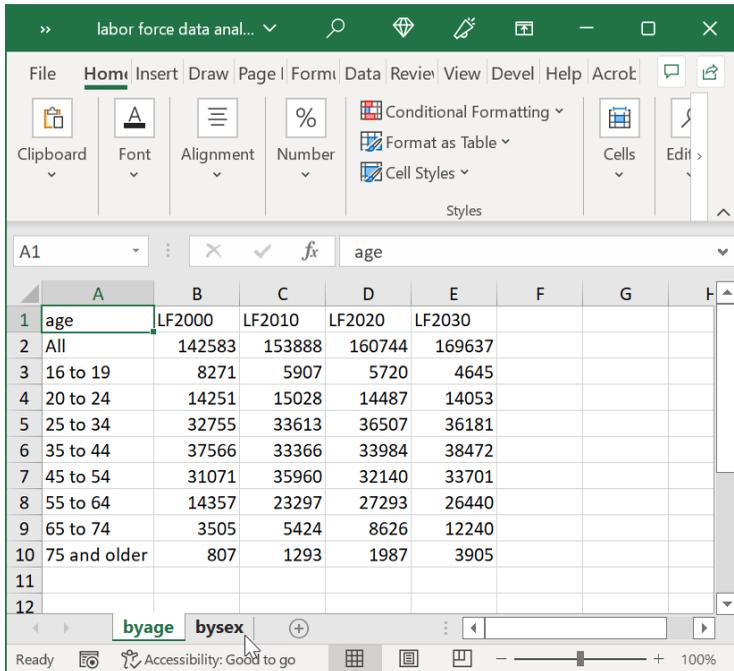


Figure 1.5: The bysex Worksheet of the Workbook labor force data analysis.xlsx created by Script 1.1



The screenshot shows an Excel spreadsheet with the following data:

age	LF2000	LF2010	LF2020	LF2030
All	142583	153888	160744	169637
16 to 19	8271	5907	5720	4645
20 to 24	14251	15028	14487	14053
25 to 34	32755	33613	36507	36181
35 to 44	37566	33366	33984	38472
45 to 54	31071	35960	32140	33701
55 to 64	14357	23297	27293	26440
65 to 74	3505	5424	8626	12240
75 and older	807	1293	1987	3905

Figure 1.6: Worksheet "byage" created by Script 1.1

1.3.2 *Creating and Sharing Analytic Reports Programmatically*

One of the most fascinating things to me about R is that you can use it to automate the creation of your entire analysis report in Word, HTML, PDF or plain text. That is, to update your report, you will only need to change your input dataset, then rerun the R script. A good practice is usually to carefully read the report after it has been created, to ensure that the narrative matches the revised figures. You will learn how to do all this in chapter 6.

Your analysis report can also be formatted in Excel or Google Sheets, with colors and fancy fonts using the R language. This issue will be discussed in details in chapters 7, 8 and 9.

To give you a flavor of what you are going to learn, I created a short US labor force analysis report in MS Word, which you can download from the link

1.3. Using R to Automate Spreadsheets

<https://bit.ly/3pn5VkI>. It contains a basic analysis of the US labor force data shown in Table 1.2 . This Word document contains 2 data tables, one barplot and a narrative. Figure 1.7 only shows you the introductory section of this report. You may need to download the entire report to see the remaining sections.

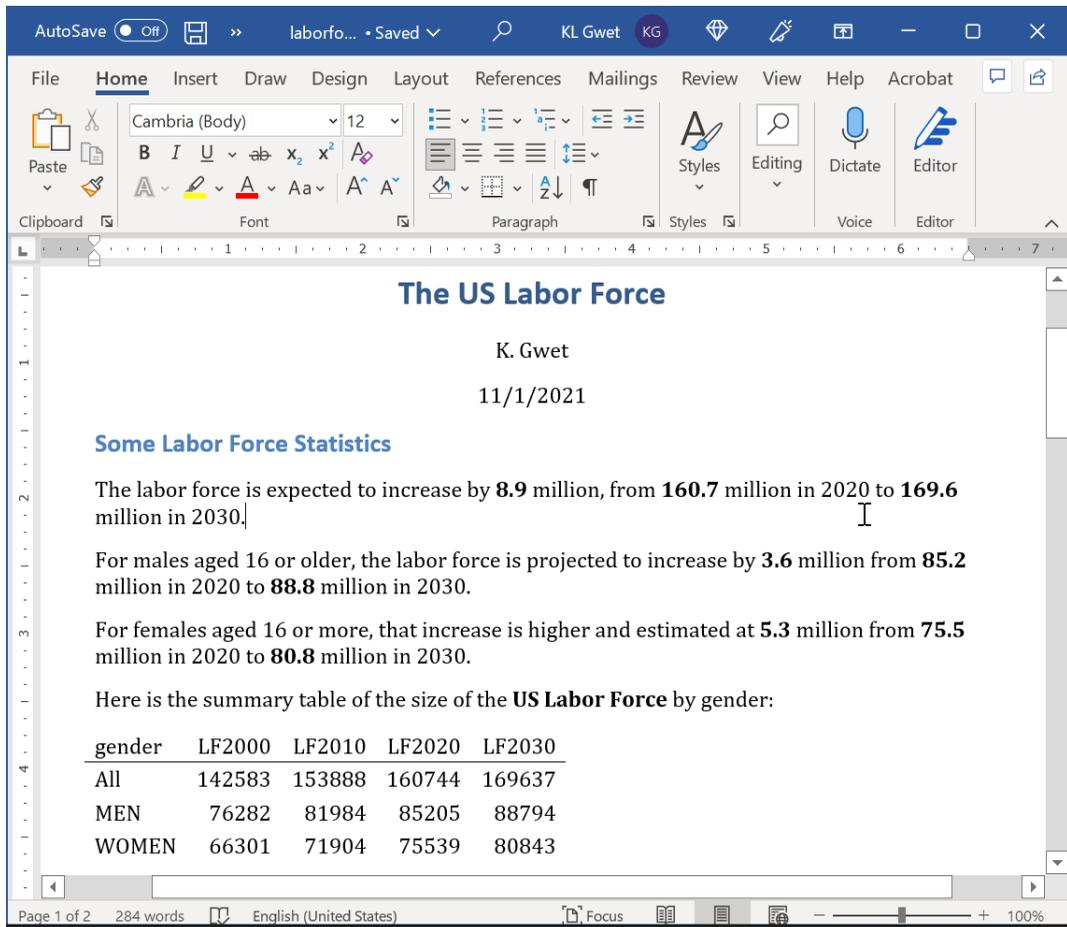


Figure 1.7: Extract of the US Labor Force Analysis Report in Word

All statistics seen in that report, including the numbers in boldface used in the narrative, were calculated by an R script. The only input data needed to generate this report is the dataset of Table 1.2 . If you modify it, then

you can generate an updated Word document with a single mouse click. To see what the R script used to do this work looks like, follow the link <https://bit.ly/3w5d027>. Figure 1.8 shows you the portion of this R code used to create the first few paragraphs of the Word report. This is a special R script file that is known as an RMarkdown document. It is a combination of text and R commands. After you read this book, I expect you to be able to use R in similar tasks.

```

19 by.gender.tot <- (Tra.Iforce %>%
20   summarise(across(where(is.numeric),sum)) %>%
21   mutate(gender="All"))[,c(5,1:4)]
22 by.gender <- rbind(by.gender.tot,by.gender)
23
24
25 ## Some Labor Force Statistics
26
27 The labor force is expected to increase by **`r
round(by.gender.tot$LF2030/1000 - by.gender.tot$LF2020/1000,1)`** million,
from **`r round(by.gender.tot$LF2020/1000,1)`** million in 2020 to **`r
round(by.gender.tot$LF2030/1000,1)`** million in 2030.
28
29 For males aged 16 or older, the labor force is projected to increase by
30 **`r round((by.gender$LF2030[2]-by.gender$LF2020[2])/1000,1)`** million from
**`r round(by.gender$LF2020[2]/1000,1)`** million in 2020 to **`r
round(by.gender$LF2030[2] / 1000,1)`** million in 2030.
31
32 For females aged 16 or more, that increase is higher and estimated at **`r
round((by.gender$LF2030[3]-by.gender$LF2020[3])/1000,1)`** million from **`r
round(by.gender$LF2020[3]/1000,1)`** million in 2020 to **`r
round(by.gender$LF2030[3] / 1000,1)`** million in 2030.
33
34 Here is the summary table of the size of the **US Labor Force** by gender:

```

Figure 1.8: Extract of the US Labor Force Analysis Report in Word

1.4. Concluding Remarks

1.4 Concluding Remarks

I have used both R and Excel for close to 2 decades⁵, and will continue to do so. In this chapter I reviewed many advantages that justify the use of Excel. I also pointed out many Excel's weaknesses that made me consider alternative tools such as R. I showed that R can be an effective language for automating many time-consuming Excel tasks.

R is not part of Excel, but can read an Excel data file, analyze it within the R powerful computing environment and report the results back to Excel with all the formatting required. The end user of the final Excel workbook will have no way of knowing that the work was not done in Excel. Chapters 7 and 8 provide a detailed account of this impressive R capability.

Microsoft has introduced **Visual Basic for Application (VBA)**, **Power Query**, and **Power Pivot** to improve Excel's analytical capability. Developing a working VBA solution is slower than developing an R solution. It is because R has a far more diverse analytical toolset than Excel, and has the support of larger community of users who can assist you with technical issues. **Power Query**, and **Power Pivot** have expanded Excel's capability considerably. However, it remains difficult for others to look at your work and for yourself to maintain your solutions over time.

In chapter 2, you will learn how to set up the R environment on your computer. Chapter 3 is devoted to the study of R datasets. Here is where you will learn how to organize your data once it is transferred to R. Chapter 4 is the place where you will learn many of the powerful tools R offers for analyzing your data. Data visualization is discussed in chapter 5. You will learn how to use R for creating many types of charts commonly-used by Excel analysts. When it comes to data visualization, R is likely the best option available. The important topic of programmatic report is discussed in chapter 6. Chapters 7,

⁵Actually, as a graduate student in the early nineties, I was already a heavy user of S, the language R is based on.

8 focus on the interaction between R and Excel, whereas interaction between with Google Sheets is discussed in chapter 9. The importance of version control with GitHub is growing among modern-day data scientists, and chapter 10 is devoted to it.