

R Introduction Part I

Dr. Robert Buscaglia

October 22, 2018

Getting Started

R

R is a free object-oriented “programming” language. It provides an environment for statistical computing and graphing. R is available on a wide range of platforms. It can be installed, for FREE, from the following link.

<https://cran.cnr.berkeley.edu/>

R Studio

RStudio provides a graphical user interface (GUI) for helping to navigate the R language. RStudio has also been developed as an open source program!

<https://www.rstudio.com/products/rstudio/download/>

It provides a useful (and for some of us necessary) overlay for working with R. It allows you to visualize your work as you go. When you create an object, it gives you a GUI to see that object. Although it can be slower for large computations, it can be invaluable for developing in-depth and useful code.

R Markdown

R Markdown (RMD) is an R package that can be used to create documents, presentations, websites, and many other forms of viewable results. RMD allows the user to incorporate code and word processing seamlessly, resulting in high quality documents with easy to follow flow; from figures, to simulations, to results!

This entire document was created in RMD. I use RMD almost exclusively for preparing lectures, presentation of results for research, and summarizing my work as I go!

If using a personal computer, RMD must first be installed using the R (or R studio) interface.

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

Notice that code was added seamlessly to the presentation. RMD also allows you to present figures and tables; anything you can produce in R can be displayed through RMD, and turned into a presentation or publication quality PDF.

R Basics

There are many sources available for learning how to use R. This includes digital and hard-copy books, as well as online references. There are even packages to help you learn R through R! Provided below are a list of those I have found to be the most useful.

Introductory Textbooks

- The Art of R Programming: A Tour of Statistical Software Design, Norman Matloff
- R For Dummies, Andrie de Vries
- An Introduction to R, William N. Venables

Statistical Learning in R

- An Introduction to Statistical Learning with Applications in R
 - Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
 - <https://www-bcf.usc.edu/~gareth/ISL/>

Online Learning Tools

- DataCamp
 - <https://www.datacamp.com/>
 - Free but does requires you to create an account!
 - Good for programming beyond R.

Learn R in R!

- Package “swirl”
 - Allows you to load learning modules directly in R studio console.

NAU Textbook

- A Sufficient Introduction to R, Derek L. Sonderegger
 - <https://dereksonderegger.github.io/570L/>

The Ugly Truth

- Learning to program in R, like any other language, takes time and practice.
- The more you use it, the better you get!

A First Demonstration

This demonstration will be presented here as well as within R Studio and R Markdown.

Scripts vs. Rmarkdown

Two file types that are important when working with R and RMD.

.R File - script file. These allow you to save your code as you go. Running lines of code from a script file will be displayed in the console window. You can work with R like a calculator, and save the computations for later reference!

.RMD File - R markdown file. This is where we can integrate text, code, and latex to create presentations of results with added annotations. However, sometimes developing code within RMD is not very efficient, and it is better to work in a .R file. Once code has been perfected, it can be moved to a .RMD file and placed within a code block.

R as a calculator

R can be used to do every day calculations. We can try simple computations.

```
12 + 12
```

```
## [1] 24
```

```
sqrt(100)
```

```
## [1] 10
```

```
log(10)/exp(10)
```

```
## [1] 0.0001045372
```

Object Oriented

A main feature of using R is building “objects” or variables through assignment. Once we create assignments, we can reuse them within R.

```
x<-2*pi
```

```
x
```

```
## [1] 6.283185
```

```
cos(x)
```

```
## [1] 1
```

Simple built-in functions

R offers a host of built-in simple functions to allow you to work out common mathematics and statistics problems.

If you are ever unsure of what function you may need, R studio provides easy access to help documentation. If you are ever unsure what a function does, you can always use the “?” to ask it to give you information.

```
?mean
```

Let's view some other useful commands:

```
x.2<-seq(1, 2, 0.1)
```

```
x.2
```

```
## [1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0
```

```
mean(x.2)
```

```
## [1] 1.5
```

```
median(x.2)
```

```
## [1] 1.5
```

```
quantile(x.2)
```

```
## 0% 25% 50% 75% 100%
```

```
## 1.00 1.25 1.50 1.75 2.00
```

```
summary(x.2)
```

```
## Min. 1st Qu. Median Mean 3rd Qu. Max.
```

```
## 1.00 1.25 1.50 1.50 1.75 2.00
```

If you want to double check something using R, there is likely a function to help. *Google* is your friend!

Data Types and Storage

The type of object we are creating is important as well as the method we use to store it. There are many data types within R. A few include:

- Numeric
- Logical
- Strings

There are also many storage types:

- Scalar

```
x.3 <- 3
```

```
x.3
```

```
## [1] 3
```

- Vector (Concatenation is important!)

```
x.4 <- c(1,2,3,4)
x.4
```

```
## [1] 1 2 3 4
```

```
x.5<-seq(1, 2, 0.25)
x.5
```

```
## [1] 1.00 1.25 1.50 1.75 2.00
```

```
x.6<-1:5
x.6
```

```
## [1] 1 2 3 4 5
```

```
x.7<-x.6*2
x.7
```

```
## [1] 2 4 6 8 10
```

- Matrices

```
x.8<-cbind(x.5, x.6, x.7)
x.8
```

```
##      x.5 x.6 x.7
## [1,] 1.00  1  2
## [2,] 1.25  2  4
## [3,] 1.50  3  6
## [4,] 1.75  4  8
## [5,] 2.00  5 10
```

```
x.9<-rbind(x.5, x.6, x.7)
x.9
```

```
##      [,1] [,2] [,3] [,4] [,5]
## x.5    1 1.25  1.5 1.75    2
## x.6    1 2.00  3.0 4.00    5
## x.7    2 4.00  6.0 8.00   10
```

```
x.8%*%x.9
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  6.00 11.2500 16.500 21.7500 27.0
## [2,] 11.25 21.5625 31.875 42.1875 52.5
## [3,] 16.50 31.8750 47.250 62.6250 78.0
## [4,] 21.75 42.1875 62.625 83.0625 103.5
## [5,] 27.00 52.5000 78.000 103.5000 129.0
```

- Data Frames

Data frames are one of the most versital parts of R! They allow us to store any data type within a structure that looks like a matrix. We can store strings, with logicals, with numerical data. We can even have lists of lists stored within a data frame. Data frames are an important structure within R and can take some time to get used to. Here is an example.

```
Grades <- data.frame(  
  Name = c('Bob', 'Jeff', 'Mary', 'Valerie'),  
  Exam.1 = c(90, 75, 92, 85),  
  Exam.2 = c(87, 71, 95, 81)  
)  
Grades
```

```
##      Name Exam.1 Exam.2  
## 1     Bob     90     87  
## 2     Jeff     75     71  
## 3     Mary     92     95  
## 4 Valerie     85     81
```

Notice that it stores both strings and numerical values. We can then call values using the variable names assigned.

```
Grades$Name
```

```
## [1] Bob      Jeff      Mary      Valerie  
## Levels: Bob Jeff Mary Valerie
```

```
Grades$Exam.1
```

```
## [1] 90 75 92 85
```

We will dive deeper into data frames next time.

Packages

One of the best properties of R is that it is free! This means that many people are always developing new content to be used within R. This content is commonly dispersed within what are known as Packages. Packages can be downloaded and installed manually, but can also be installed through R commands. We use the command `install.packages()` where within the `()` we enter the name of the package we are interested in. For example, we will want to use “ggplot2” to make nice graphics. First, we must install “ggplot2”.

```
install.packages("ggplot2")
```

It is best to install packages from the console window or within a script. Once you have installed a package, you have it permanently with that version of R. If you update R, or R Studio, you will typically have to reinstall packages. If you change computers, you also will have to reinstall packages!

Once we have a package installed, most (but not all), packages must be loaded before they can be used. We load packages we want to use with the `library()` function.

```
library(ggplot2)
```

Some packages include unique data sets that can be used for either learning, training, or testing purposes. One large compendium of data sets is the “Lock5Data” package. Below we load the package (after we have installed it) and call from it a data set known as BaseballHits. There are many ways we can do this, here are two:

```
library(Lock5Data) ### FIRST LOAD LIBRARY
data(BaseballHits) ### THEN THE DATA IS AVAILABLE
data(BaseballHits, package="Lock5Data") ### OR: LOAD DATA BY INDICATING THE PACKAGE
```

Once we have the package and data loaded, we can pull information from that data set. The data has actually been pulled in as a data frame!

```
head(BaseballHits, 5)
```

```
##   Team League Wins Runs Hits Doubles Triples HomeRuns RBI StolenBases
## 1  ARI     NL   64  615 1379    259     47    118  573         86
## 2  ATL     NL   79  573 1316    240     22    123  545         95
## 3  BAL     AL   96  705 1434    264     16    211  681         44
## 4  BOS     AL   71  634 1355    282     20    123  601         63
## 5  CHC     NL   73  614 1315    270     31    157  590         65
##   CaughtStealing Walks Strikeouts BattingAvg
## 1                33   398        1165     0.248
## 2                33   472        1369     0.241
## 3                20   401        1285     0.256
## 4                25   535        1337     0.244
## 5                40   442        1477     0.239
```

```
BaseballHits$Hits
```

```
## [1] 1379 1316 1434 1355 1315 1400 1282 1411 1551 1557 1317 1456 1464 1476
## [15] 1399 1366 1412 1306 1349 1354 1356 1436 1199 1328 1407 1371 1361 1400
## [29] 1435 1403
```

Using R Markdown

R Markdown allows us to integrate everything we've learned above, but to also include text, latex, graphs, tables. I make lectures, presentations, prepare results, even this document is RMD!

To create a RMD file, you must first open one in RStudio (if you are using base R, you will have to do this from command line). When you open the file, you are asked to give it a Title, the Author, and choose an output type.

You can then integrate code and text (and latex, etc. . .) with easy work flow. Most of the work is typed in to the text blocks, much like using a text editor (Word for example). We can also create code blocks for running and displaying output. Using the Insert > R on the top right of R studio will add an R code block for you. You can also type it in manually. To ensure the code is displayed. It is important to also include the option, `{... , echo=TRUE}`.

Here is an image of a basic RMD setup, showing code blocks and text blocks (taking from online).

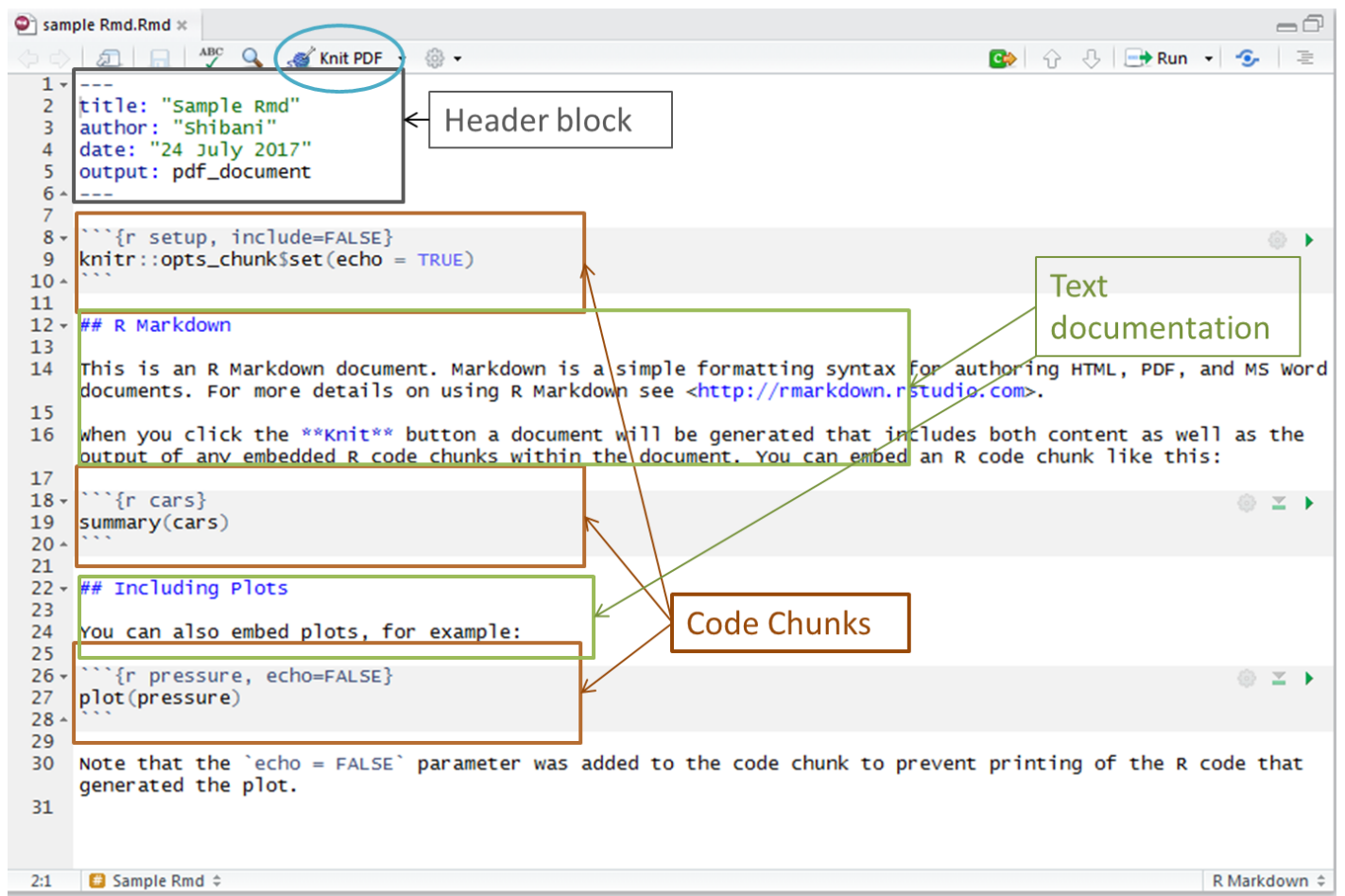


Figure 1: RMD Code and Text Block Flow

For the assignment this week, we want to create an RMD file with some basic computations. Once you have created code and text chunks, you must “KNIT” the document. The ADEL 222 computers are set up well to do this. If trying to use a personal computer, be warned that you may encounter may issues when attempting to KNIT a PDF.

Assignment Week 1

This assignment can be completed in the Math and Stats Computer lab (ADEL 222). If you choose to work on a personal computer, be prepared that R Markdown may not compile PDFs due to issues with Latex interfaces. Although work can be done on a personal computer, if issues with creating a PDF are encountered, you should use ADEL 222 to finalize your assignment.

The goal of the assignment is to create a simple R Markdown file with some basic work. You should work within the R studio GUI to ease learning the R environment. All code blocks must use the option `{..., echo=TRUE}`, so that both the code and the output will be shown to me.

1. Install the package “rmarkdown”. Install the package “Lock5Data”. Install the package “ggplot2”
 - Steps 1 should *not* be done in the RMD file.
2. Create an RMD file (FILE > NEW FILE > R MARKDOWN...).
 - The RMD file should be title as : “MAT 587 RMD Assignment 1”.
 - Author should include your full name.
 - Select the option for generating a PDF.
 - You will have to submit a .PDF generated directly from RMD.
 - Do not create a .PDF by converting a .DOCX into a .PDF. I will know the difference!
 - *Do not save the original file as .pdf, this will cause the file to not work!*

The instructions for steps 4 - 9 should be shown in text blocks above the code blocks

3. Create a code block loading the package “Lock5Data”.
4. Create a code block loading from the “Lock5Data” package the data set “ChickWeight”.
5. Create a code block to display the first 10 entires from ChickWeight using the following code:

```
head(ChickWeight, 10)
```

6. Create a code block loading the package “ggplot2”
7. Create a code black that will graph of the weights vs. time for each Diet within the ChickWeight dataset using the following:

```
ggplot(ChickWeight, aes(x=Time, y=weight, colour=Diet)) +  
  geom_point(alpha=.3) +  
  geom_smooth(alpha=.2, size=1) +  
  ggtitle("Fitted growth curve per diet")
```

8. Comment within a text block on which diet seems to produce the largest weight at Time = 20.
9. Create a code block generating a sequence of number from 1 to 100 with length 19.
 - Assign this sequence of numbers to the variable *My.Seq*.
 - Be sure to display the results as output.
10. Create a code block that displays the 10th value of *My.Seq*.
11. Create a code block that calculates the mean of *My.Seq*.
12. Create a code block that calculates the median of *My.Seq*.

Print the .PDF and submit in class next week! I encourage you to stop by my lab hours T/Th 2:20 - 3:10 PM if you have questions about R!