Week_2_IntroToRmd-Map_of_RStudio

Seema Plaisier

07/01/22

Rmd format

This is an R Markdown document. Markdown is a simple formatting syntax that allows you to run and print R code into HTML, PDF, and MS Word documents using a tool called Knitr. For more details on using R Markdown see http://rmarkdown.rstudio.com. It contains descriptive text with specific symbols that correspond to text formatting options and R code in sensible pieces (called "chunks"). You can run each chunk individually in RStudio to develop and correct code and then run/print the whole Rmd into a report.

Each Rmd has a header at the top. This can be used to set the author of the document and the date it was written as well as specify the output format: html_document, pdf_document, word_document are the most commonly used, see https://rmarkdown.rstudio.com/lesson-9.html for the full list.

In this Rmd, the 'date' field of the header is set to fill automatically with the date that the report is being run. This is meant to serve as a time stamp to let people know when your report was run last to help in case there are multiple versions of the code out there run at different times in different stages of development.

You will see in this document that there are headings for sections of the code marked with "##" at the beginning of the lines between the code. These allow the creation of quick-link table of contents for the report that you can print out (explained more below). The principle heading is marked with a "#" and using more hashtags indicates secondary subheadings. So if you have one heading marked with "#" and the next marked "##", the "##" heading will be accessible as a pull-down in the table of contents under the first section marked with a "#". You can play around with this if you are interested; this tutorial uses "##" for all the headings so the size of the header text is a smaller size and all headings are at equal level.

Code and comments

This is the first chunk in an Rmd file. This chunk tells R to print out useful messages as it runs through the chunks of code you write. You will see them in the Render tab down below as reports are printed. Each chunk is surrounded by two sets of three single quotes (the one that goes with the ~ symbol under the escape key of your keyboard). Each chunk has a header which is surrounded in curly braces, starts with 'r' indicating that this is R code, a name which comes just after the r, and optional parameters you can pass in. You won't need to know much more than that for this course, but if you are curious, see here: https://rmarkdown.rstudio.com/lesson-3.html

An example of a simple R chunk:

```
print("Refreshing my memory with R")
```

```
## [1] "Refreshing my memory with R"
```

If you run specific lines of code, the output will show in the Console and Plots tables (below and right). If you run the whole chunk it will show up below the code in the Rmd tab.

Inside your chunks, you are writing R code and can describe that code using comments. Comments are lines that have a hashtag symbol at the beginning. These lines are not interpreted as code and basically ignored by the R engine. To execute lines of code, you can copy and paste them into the Console, or selected using the Code menu option Run Selected Line(s) (or matching key commands that it will tell you) or use the Run button (see pull-down for specific options) both of which do exactly that.

Code consists of creating variables and passing those variables into functions. Variable names have to be a single word (no spaces or punctuation marks). They can be set to have values using the arrow assignment operator '<-' or the equal sign '='. You will see both on the internet; code from this class primarily uses the equal sign.

```
# this is a comment, indicated by a # symbol at the beginning of the line
# comments are not interpreted as code
# they are used to help explain code to others as they go through it

# below this line is real code
print ("This is code-- print is a function and this message is passed into it")
```

```
## [1] "This is code-- print is a function and this message is passed into it"
```

```
# we can use variables to store information
# here the variable called print_this_line is created and set to store specific test
# variables in R can be set using the arrow operator '<-' or an equal sign '='
print_this_line = "Hello World!"

# the value of this variable can then be passed into a function
# here we use the function 'print' that comes with R by default
print (print_this_line)
```

```
## [1] "Hello World!"
```

```
# you can change the value of variables
print_this_line = "And now print this!"

# and then pass those into a function
print (print_this_line)
```

```
## [1] "And now print this!"
```

```
# this variable is a data frame, a type of variable you can set in R
# if you run this line, you should see a variable called emp.data in your Environment on the upper righ
emp.data <- data.frame(emp_id = c (1:5), emp_name = c("Rick","Dan","Michelle","Ryan","Gary"), salary = 
# double click the variable so you can look at it in a new tab above
```

Things to notice in RStudio

Look around at your coding environment in RStudio.

Notice that each line is numbered to the left. These line numbers are referenced in error messages that appear in the Console or Render tabs if something goes wrong and can be used to help figure out what happened.

Notice that this code is opened in a tab. You can open multiple Rmd files so you can easily copy code from file to file for similar applications.

The Console tab below is where you will see code that is run. The Render tab is used for this purpose if you run the code during report generation using Knitr. You can write code directly into the console to test it out and then copy and paste it into your Rmd when you have it outputting what you want in the way you like.

To the right, you should see an Environment tab. This will contain all the variables you store data into. Everything in the Data section of the Environment should be viewable by double clicking; the contents will be opened in a tab along side your Rmd tab.

This same section on the right contains a History tab which lists all the commands entered in the Console, as individual commands or by running a chunk of code. Click on any command and you can use the To **Console** button to send it to the Console tab to press enter and run or the **To Source** button to send it to wherever your cursor is currently in your Rmd.

Notice below that is a panel where you can view the Files in your current working directory (see below for how to change that). Plots is used to display any plots generated by your code (they will be displayed in the report if run with Knitr). Packages lists all the packages currently installed in your environment that you can load with the 'library' function. Help can be used to search for specific functions.

Knitting a report

You should see a **Knit** button in R Studio. If you don't, copy and paste this command into the Console or press the Play button on the right side which does exactly that. 'eval' is set to FALSE because we will only need to do this once (not regularly) and 'include' is set to true so the code shows up on the report. After Knitr is installed, you should be able to click on the Knit button in RStudio and find a report document in the same directory as this code. The document generated should include both descriptive content as well as the output of any embedded R code chunks within the document.

```
# if you don't see a Knit button above, run this line
install.packages('knitr', dependencies = TRUE)

# if you get an error before your pdf is made,
# copy this line into your Console and press enter to run
# answer yes ('y') to install any dependencies
tinytex::install_tinytex()  # this installs drivers to write to a PDF

# try to Knit to PDF and find your PDF file in the working directory

# after the PDF is completed, RServer will automatically try to open
# the pdf for you to view

# you may need to allow pop-ups in the browser window you are using
# to run RStudio Server
```

List all the packages used for future reference

It's good practice to include this chunk at the end of all of your Rmd files. It lists out all the packages used in when this code was run including all the version numbers. This type of information should be included

in any publications or technical reports to ensure reproducibility. It can explain differences in output when installing packages after some time has passed and can be used to address problems involving specific versions of packages are listed as dependencies for other packages.

```
sessionInfo()
```

```
## R version 4.2.0 (2022-04-22 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.2.0  magrittr_2.0.3  fastmap_1.1.0   cli_3.3.0
##  [5] tools_4.2.0     htmltools_0.5.2 rstudioapi_0.13 yaml_2.3.5
##  [9] stringi_1.7.6   rmarkdown_2.14  knitr_1.39      stringr_1.4.0
## [13] xfun_0.31       digest_0.6.29   rlang_1.0.2     evaluate_0.15
```