Lab 0: Introduction to R Markdown

Dave Harrington, Julie Vu

May 2018

This is an R Markdown (Rmd) document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. It was designed to simplify creating HTML documents, but the labs for this text are written to generate PDF documents. Before continuing, be sure to install a full TeX distribution from https://www.latex-project.org/get/.

The content of an R Markdown document is created in the R Studio script editor. Formatting commands in the text are converted to a PDF document when you click on the *Knit* button, located on the toolbar at the top of the script editor.

In R Studio, the menu item *File > New File > R Markdown...* choice produces a dialog box for output type. To create a new document, select *Document* from the left side of the dialog box, enter a name and title, choose PDF, and select OK. A template will open in the script editor, with file extension .Rmd. Templates will be provided for all the lab exercises, and you will start by simply opening the template files.

Let's get started.

The first six lines of this file are referred to as the **header**. R Markdown is very fussy about the form of the header - the three dashed lines above and below lines 2-5 must appear exactly as in this document, and there must be a blank space between the colon and the descriptive text. Also, the title, author, and date fields must be kept within double quotes. The output line specifies the output format as PDF.

1. First, rename this file to include your first initial and last name – e.g., 01_intro_to_Rmd_dph.Rmd. (Use *File > Save As...*)
2. In this document, edit the header to include your name and today's date. Click *Knit PDF*. This should produce a PDF file located in the same folder as the Rmd file, with a name like 01_intro_to_Rmd_dph.pdf. Note that the file name for a PDF created from an Rmd document will be the same, except with a different file extension. The file name and title of the document, however, can be different.

Plain text is prepared in paragraphs, as in the first part of this document. *Text* enclosed in asterisks is *italicized* in the PDF output. **Text** enclosed in double asterisks appears in **bold font**. There must be no space between the asterisks and the enclosed text.

3. Write a brief paragraph describing previous coursework in statistics (if any) and share your motivation for learning statistics with R. *Knit* the document. Note that each time you *knit* the document, the output overwrites the previous version.

Your paragraph...

Bulleted lists are produced using the formatting syntax:

- Item 1
- Item 2
    - Item 2a
    - Item 2b

The list must be preceded by a blank line, and 4 spaces should be used before sub-items.

4. Write a bulleted list giving your year of graduation, your field of study, and the country you are from. Under the entry for your country, prepare subitems with the name of your state and city. *Knit* the document and inspect it to make sure the PDF is correctly produced.

Your bulleted list…

Additional formatting commands will be introduced gradually throughout the rest of the labs.

5. To start a new page in the PDF document, enter the text ‘newpage’ preceded by a backslash, as in… (new page coming in the PDF!)

Using R with R Markdown

The real power of R Markdown is that it allows for R programs to be included in the Rmd file, with both the program and its output automatically being produced in the PDF document.

R programs in an Rmd file are located in **code chunks**. You can embed an R code chunk by either typing the three apostrophes followed by an "r" enclosed in braces, then the additional three apostrophes to close the chunk, or simply press the *Insert Chunk* button from the *Chunks* menu on the far right of the toolbar on top of the script editor.

6. Datasets for the labs are contained within the *oibiostat* package. To install the *oibiostat* package, run the code in the following chunk. The package only needs to be installed once.

```
install.packages("devtools")
```

```
##
## The downloaded binary packages are in
##  /var/folders/lj/6hkky8yj1bd9p35w3hv5gv9r0000gn/T//RtmpXtOcce/downloaded_packages
```

```
devtools::install_github("dave-harrington/eventtimedata")
```

```
## Downloading GitHub repo dave-harrington/eventtimedata@master
## from URL https://api.github.com/repos/dave-harrington/eventtimedata/zipball/master
```

```
## Installation failed: Not Found (404)
```

To view which datasets are included in the package, go to the "Packages" tab and scroll down to *eventtimedata*; clicking the blue hyperlink opens the documentation page that lists the datasets included in the package.

7. The following code chunks use data included in the package. To load the package, use the `library` command. This command only needs to be used once in a document, in the first R chunk that requires data from the package. The `data()` command can be used to load a particular dataset from the package; once the command is run, the name of the dataset will appear in the Environment tab.

```
library(eventtimedata)   #loads the package
data(rossi)            #loads the rossi data
table(rossi$arrest)
```

```
##
##   0   1
## 318 114
```

```
mean(rossi$age)
```

```
## [1] 24.597
```

When an analysis requires a function that is contained in a package, the package must be loaded with a library statement as well. The package most often used in this course is `survival`. This code chunk computes survival function estimates for the `rossi` data. The syntax for `survfit()` will be explained in lecture.

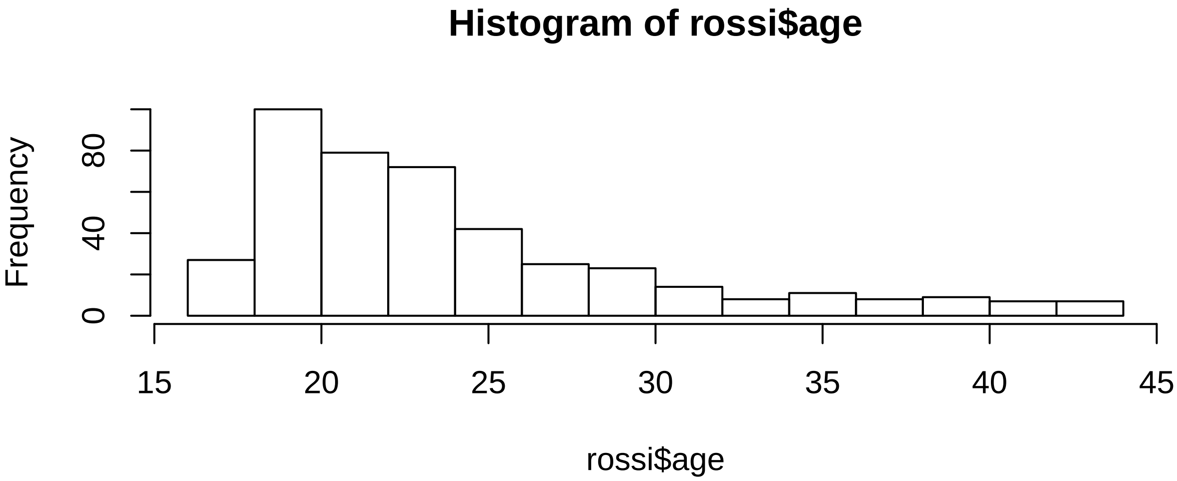
```
library(eventtimedata)   #loads the package
library(survival)        # loads the survival package
data(rossi)            #loads the frog dataset
survfit(Surv(week, arrest) ~ 1, data = rossi)
```

```
## Call: survfit(formula = Surv(week, arrest) ~ 1, data = rossi)
##
##       n  events  median 0.95LCL 0.95UCL
##     432     114      NA      NA      NA
```

8. It is also possible to view the output of a code chunk without having to *Knit* the entire document.

- To run a single line, place your cursor on the line you want to run and press Ctrl/Cmd + Enter. Try this out with the lines in the following chunk. The output appears directly below the code. To clear the output, press the X in the upper right hand corner of the preview.
- To run an entire chunk, place your cursor within the chunk and press Ctrl/Cmd + Shift + Enter. Try this out in the following chunk. The output is now accessible between the two panes.
- The Run drop-down menu in the upper right hand corner of the script editor provides other options for running chunks, such as running all chunks above or below a certain point. The gear drop-down menu, next to the *Knit* button, provides options for expanding or collapsing all output in a document.

```
hist(rossi$age)
```



```
median(rossi$age)
```

```
## [1] 23
```

Loading Datasets

9. Datasets are not always in R packages; in most cases, datasets are downloaded to the local computer from alternate sources or generated from R. The following code creates a small dataset called *sample.data* that consists of the numbers 1:9, arranged in a matrix with three rows and three columns and saves it as a file called *sample_data.Rdata*. Run the following code chunk and confirm that the dataset appears in the same folder as where this Rmd document is saved.

```
sample.data = matrix(1:9, nrow = 3, byrow = T)  #create sample dataset
save(sample.data, file = "sample_data.Rdata")   #save the file
rm(list = ls()) #clears the environment, equivalent to clicking the broom icon
```

10. To load the dataset, use the `load()` command. Note how the command uses the file name, with the .Rdata extension. In the environment pane, the dataset appears as *sample.data*, the name that the matrix was given in the previous code chunk.

```
load("sample_data.Rdata")
```

Alternatively, clicking on the name of the dataset in the Files panel will produce a dialog box asking for confirmation; choose Yes to load the dataset into *RStudio*.

R Script versus R Markdown

An R Script file can be thought of as one large code chunk; R script files are only meant to contain R commands with short plain-text comments. R Markdown files consist of both code, output, and plain text.

R script files are useful when the primary goal is to explore a dataset or test short R programs. R Markdown files are ideal for preparing documents that need to contain both explanatory text and statistical output, such as a lab report.

Congratulations! You have reached the end of Lab 0.