Working with R and R Markdown

Tayyip Altan

September 2024

Introduction

This tutorial provides the building blocks to make everyone comfortable with the environment we will be using in this course.

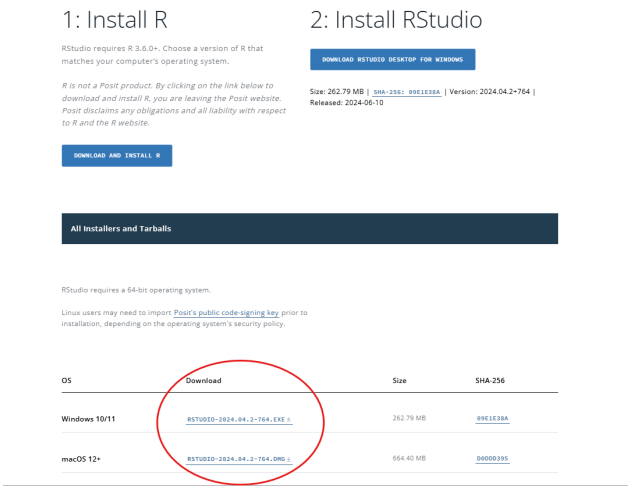
1. Getting started with the R programming language
2. R markdown (which was also used to create this document)

The R programming language

All of the tutorials and assignments in this course will make use of the R programming language. You might have already worked with R and have it installed. If not, go through the following steps:

1. Go to this page and select 'install R' for your operating system. The image below illustrates where you need to be.

2. After that, go to this page and download the free version of Rstudio. Again, select the installation for your operating system. The image on the next page illustrates where you should be looking.



It is outside the scope of this document to learn you everything about the R programming language. Rather, we will point to several resources one can use if (1) you want to get some experience in R before the course starts, or (2) if you are stuck with your R code.

- **Tutorials:**
  - **Syntax and data structures**: A beginner friendly introduction to R can be found here, which covers all the basic data structures and syntax. Additionally, Datacamp provides a comprehensive introductory course into R. Free tutorials can be found here.
  - **Plot making**: for an introduction to making plots in R with ggplot2, see the following overview.
  - **If you are stuck**, consult sources such as Google, stack overflow, or Youtube for troubleshooting assistance.

R markdown

R markdown can be used to combine (1) text, (2) R code, and (3) LaTeX to create documents. LaTeX is a typesetting system commonly used for producing scientific and technical documents due to its ability to handle formulas and complex layouts. In R Markdown, LaTeX is often used to include advanced formatting in your markdown file or to incorporate mathematical notation.

1. Installation

In order to use R markdown, you will need to install several things.

1. Usually, R markdown will be part of the R studio version you have downloaded. Just in case it is not, you can install it with the following command:

```
install.packages('rmarkdown') # install rmarkdown
```

2. If you would like to create PDF documents from R Markdown files containing LaTeX, you will need to have a LaTeX distribution installed. It is recommended to install the following custom LaTeX distribution called TinyTex.

```
install.packages('tinytex')
tinytex::install_tinytex()  # install TinyTeX
```

2. Set-up of the document

Before we can start putting text in the document, we need to go over several things in order to set it up. First, the set-up of the document. When creating a document, you define several parameters in order to make sure you get the correct output. For example, these are the parameters for the markdown file that created this document:

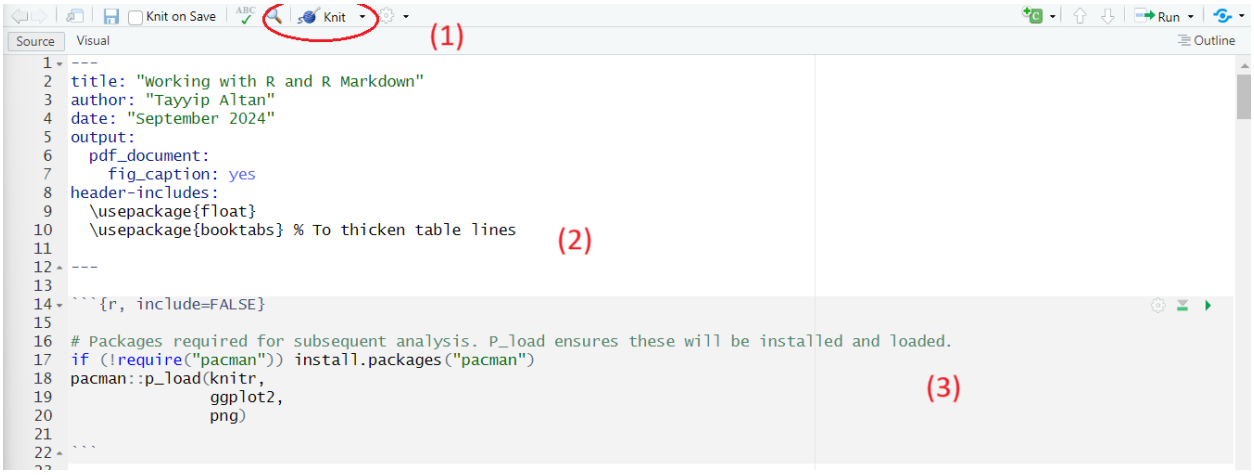
```
---
title: "Working with R and R Markdown"
author: "Tayyip Altan"
date: "September 2024"
output:
  pdf_document:
    fig_caption: yes
header-includes:
  \usepackage{float}
  \usepackage{booktabs} % To thicken table lines

---
```

The most important parameters are:

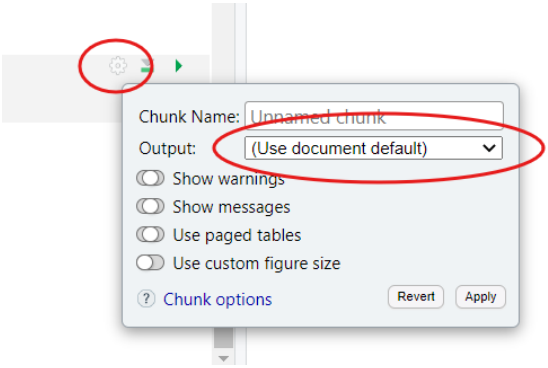
- **title, author, and date**: These are used to format the title page of the document.
- **output**: specifies what type of document you want to create. In our case, this is "pdf_document".
- **header-includes**: contains any LaTeX packages you want to add.

Once you have set the parameters, you can create the pdf output by 'knitting' the document. In the image below, the 'knit' button is circled red at (1). At (2) the parameters are defined, and at (3) we load in our libraries. We recommend using `pacman::p_load` since it automatically installs and loads any specified packages. The text and code is usually written below (3)



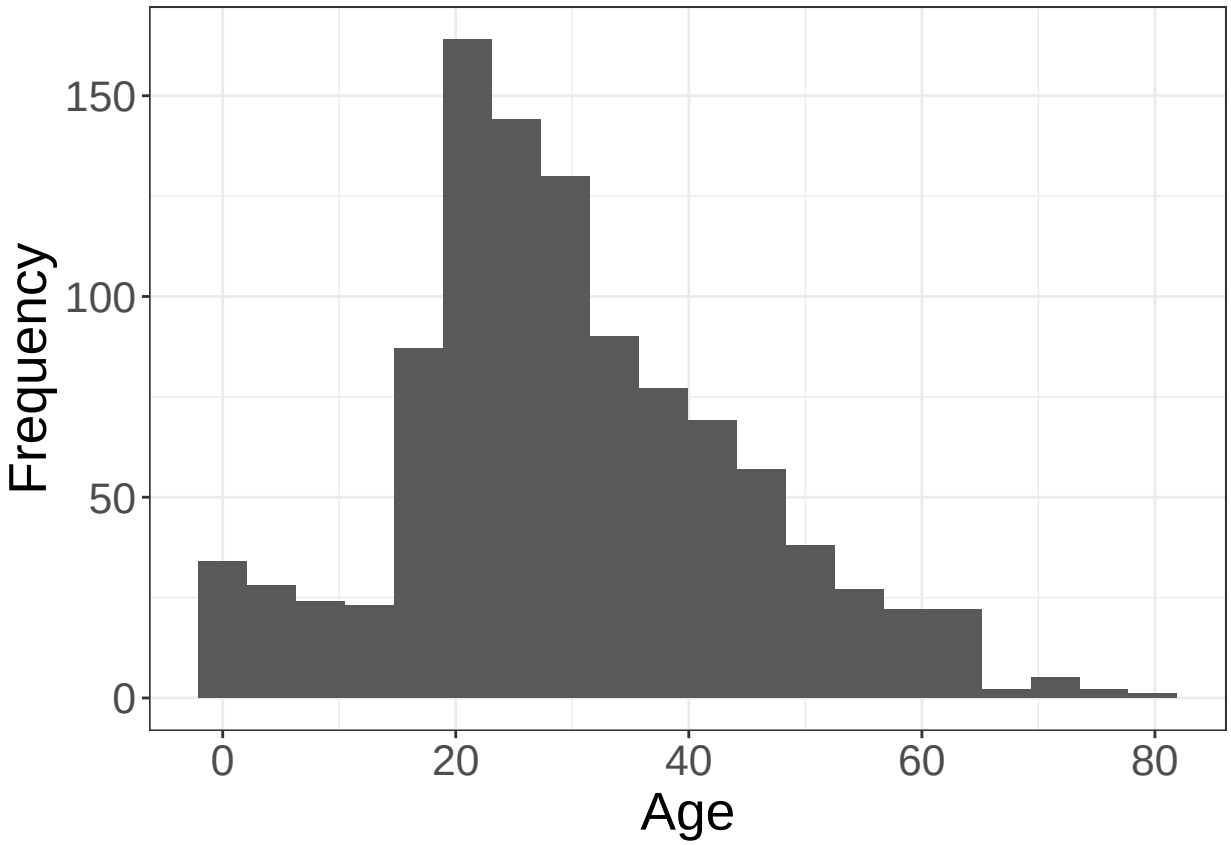
3. Writing text and code

In R markdown, one can write text the same way one writes text in LaTeX. Below is an example that loads data. The `include = FALSE` argument ensures that this code snippet is run but neither the code nor its output show in the generated pdf.

```
```{r, include = FALSE}
library(tidyverse)
```
```

Below is an overview of some of the key parameters for code chunks in R markdown.

Table 1: What each parameter does to the code chunk

Parameter	Run Code	Show Code	Show Output
eval = FALSE	No	Yes	No
echo = FALSE	Yes	No	Yes
include = FALSE	Yes	No	No

You can also use the following options for some simple changes.



4. Making plots

One of the nicest features of R markdown is that you can directly incorporate your plots into the pdf. For instance, if we want to generate a plot that is centered and takes up 50% of the page width, we can use the following parameters in our code chunk:

```
```{r, echo=FALSE, fig.pos="H", fig.align="center", out.width="50%"}
  # plot
```
```

The following is an example of how to make a plot in Rmarkdown. We use a dataset with information on passengers on the titanic.

```r
# Reads in the csv as an R dataframe
df_titanic <- read.csv('titanic.csv')

# create ggplot object of histogram of the age of passengers
age_titanic_passengers_plot <- ggplot(data = df_titanic, # dataset to work with
                                      aes(x = Age)) + # variable for x-axis is Age
                               geom_histogram(bins=20) + # create a histogram with 20 bins
                               labs(y = 'Frequency') + # name for the Y-lab
                               theme_bw() + # theme of the plot
                               theme(text = element_text(size=20)) # increase the font size

age_titanic_passengers_plot
```


150
100
50
0
Frequency
0
20
40
60
80
Age