Introduction to R
Part 1: Getting Started with R and RStudio

Aram Balagyozyan

Department of Operations and Information Management
Kania School of Management
The
University of Scranton

February 28, 2022

Outline

Outline

Introduction

- This module provides a very brief introduction to the main features of the R language.
- This course doesn't assume any prior familiarity with computer programming in general or R in particular.
- Still, this course is dedicated to applying the functionality of the R language to certain problems in Business Analytics. Therefore, throughout this course you will be expected to become familiar with and be able to use R.
- The material in this module should serve as a quick tutorial for those who are not familiar with the basics of the R language. Some other, more specific aspects of R will also appear in the subsequent modules as we proceed throughout the course.

Why R?

- R is a functional language for statistical computing. It was designed by statisticians for statisticians.Thus, R is great at data analysis and data visualization.
- R is not only free and powerful but it is also open source, meaning it is developed collaboratively; its code is open to public inspection, modification, and improvement.
- R (together with Python) has became one of the main workhorses of the Data Science community. Because of its power and open source nature, R is widely used in the field of data analytics, physical and social sciences, as well as in government, non-profits, and the private sector. As a result, there is also a large support community available to help troubleshoot problematic code (e.g on StackOverflow).

Starting with R

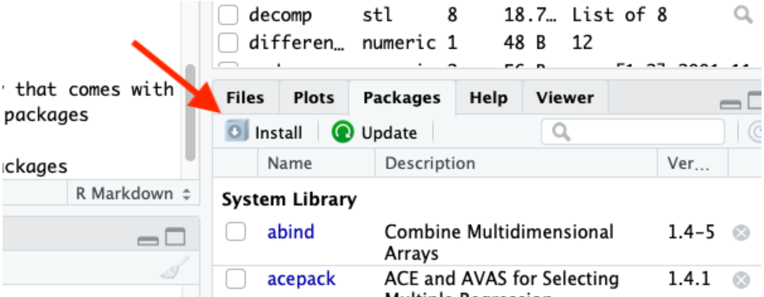
- The easiest way to install R on you system is to obtain a binary distribution from the R website where you can follow the link to the CRAN (Comprehensive R Archive Network) to obtain the binary distribution for your particular operating system.
- After downloading the binary file, install it in your system.
- Here are links to youtube videos for how to install R for Windows and how to install R for for MAC.
- Congratulations, now you have the R engine installed in your computer!

Installing RStudio

- While it is possible to interact with the R engine directly, the bare-bone R is not very user-friendly and most R users rely on an alternative integrated development environment (IDE).
- One of the most popular IDEs is RStudio is a free IDE that can be downloaded and installed for the most common setups from RStudio's website.
- Here is a link to a youtube video on how to install RStudio
- Upon installing RStudio, open it and click through `File->New File->R Script`. This will create a blank script file where you can type any R code that can be saved for later use.
- RStudio includes on the same GUI several important elements of R like
  - A Console where you can interact with R
  - A script editor where where you can write more complex programs
  - An interface where you can browse the variables you've created
  - An interface where you can browse the help pages of R and many other useful facilities.

Add-on Packages in R

- The power of R is derived first of all from the basic functionality that comes with the base install of R but also from the vast universe of add-on free packages developed and maintained by the R community.
- Each package contains a set of functions useful for specific tasks. For example, although base R contains some rudimentary plotting functions, the ggplot package contains some functions that can be used to produce beautiful elaborated plots.
- Packages can be installed from within RStudio by clicking on Install inside the Packages menu.



Add-on Packages in R

- Packages can also be downloaded programatically by running the install.packages() command.
- Suppose you want to download the package that provides functions to connect to MySQL databases. This package name is RMySQL.
- To download the package you just need to type the following command in the R prompt:

```
install.packages("RMySQL")
```

- In order to run a command, type it in the RStudio console and hit ENTER.
- One package that we might need throughout this course is **dplyr**. This package gives you access to several useful functions. So go ahead and download the package by running the following command in the RStudio console:

```
install.packages("dplyr")
```

Activating R Packages

- For a given R installation, a package must be downloaded only one time. Once a package is downloaded, it is attached to the R base package and can be activated on demand.
- In order to activate a downloaded package, run the library() command. For example, if you would like to activate the **dplyr** package, you'd need to run the following command:

```
library(dplyr)
```

- Every time an R session is reactivated (closed and reopened), the needed packages must be reactivated (although there is no need to download them more than once).
- Throughout this course, we will make use of various packages. Every time we will be making a call to a new package, I will be assuming that the package has already been installed in your computer.
- If you need to be explicit about what package a function or dataset comes from, use the special call form `package::function()`, e.g `dplyr::select()`

Updating R Packages

- Function installed.packages() reveals the packages currently installed in your computer.
- The following command can be very useful as it allows you to check whether there are newer versions of your installed packages on CRAN:

  ```
  old.packages()
  ```

- You can use the following command to update all your installed packages

  ```
  update.packages()
  ```

Outline

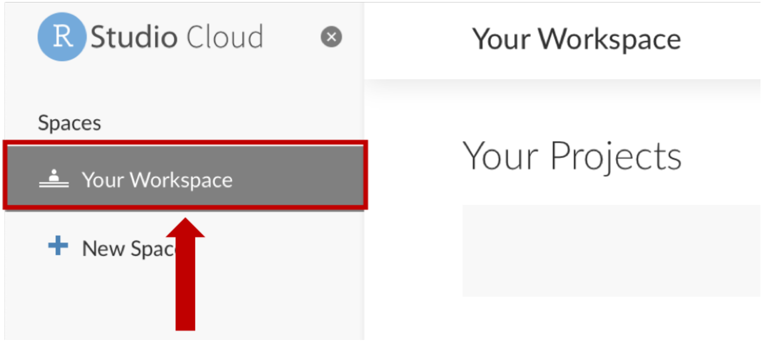
Running RStudio on the Cloud

- Although, I strongly recommend you to install and run R and RStudio on your local computer, there are two additional ways in which you can run Rstudio and execute RScript on the cloud:

1. Using Rstudio's free cloud service: https://rstudio.cloud
2. Using the University's remote lab service: https://remote.scranton.edu

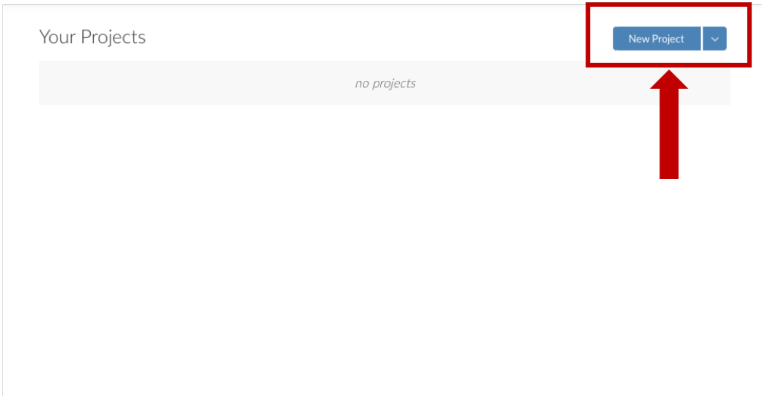
Running RStudio on rstudio.cloud

- In order to execute R code on https://rstudio.cloud, create an free account, log in, and click on `Your Workspace`



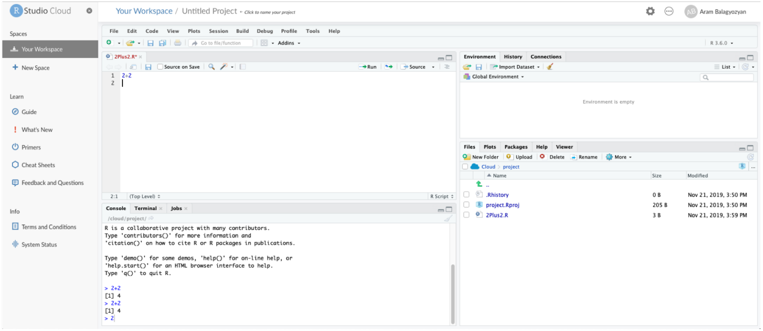
Running RStudio on rstudio.cloud

- Inside `Your Workspace` click on `New Project`.



Running RStudio on rstudio.cloud

- Once the project is created, you will see an environment that is near identical to that of the RStudio interface.



```
2+2
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 2+2
[1] 4
> 2+2
[1] 4
> 2
```

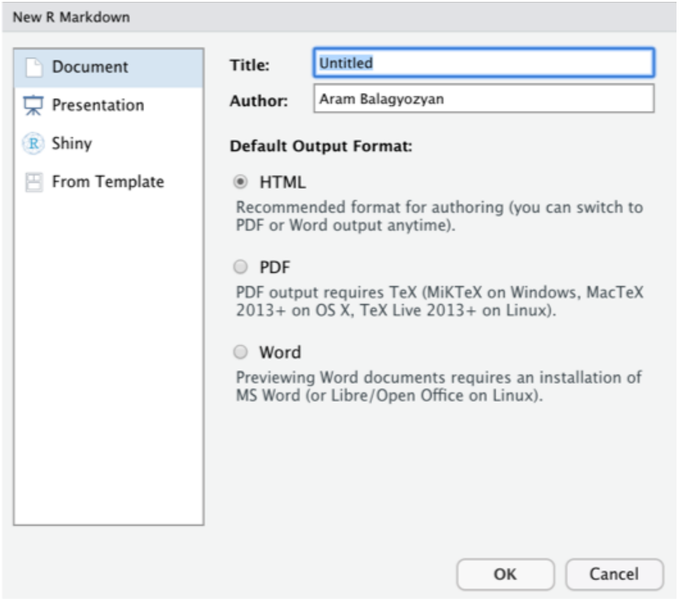
Name	Size	Modified
..		
.Rhistory	0 B	Nov 21, 2019, 3:50 PM
project.Rproj	205 B	Nov 21, 2019, 3:50 PM
2Plus2.R	3 B	Nov 21, 2019, 3:59 PM

Running RStudio on remote.scranton.edu

- You can also run RStudio on the University's remote lab.
- Log in to https://remote.scranton.edu using your Scranton credentials (note: it's not obvious but once you enter your scranton user name and password, the system will wait until you authenticate your log-in attempt using DUO. So go ahead and use DUO to authenticate your log-in attempt).
- Click on `BYODKSOM`
- Click on the `Window` icon on the bottom left corner, scroll down to and expand the `Rstudio` folder, and click on `Rstudio`.
- In order to initiate a new script file, click through `File->New File->Rscript`
- The supplementary material section of this week's learning resources contain two additional documents with more detailed description on how to access R and RStudio and save files on remote.scranton.edu.

Outline

Using R Markdown to Combine Documents with Code and Output



New R Markdown

Title: Untitled

Author: Aram Balagyozyan

Default Output Format:

[✓ HTML]

Recommended format for authoring (you can switch to PDF or Word output anytime).

PDF

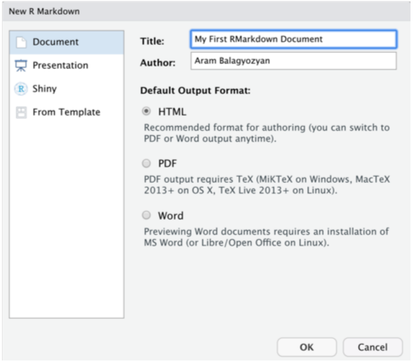
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

Word

Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

Using R Markdown to Combine Documents with Code and Code Output

- In the window that pops up, select `Document` on the left and enter the Title of the document and select HTML for the output format (as PDF requires additional software).



Title: My First RMarkdown Document

Author: Aram Balagyozyan

Default Output Format:

(●) HTML
Recommended format for authoring (you can switch to PDF or Word output anytime).

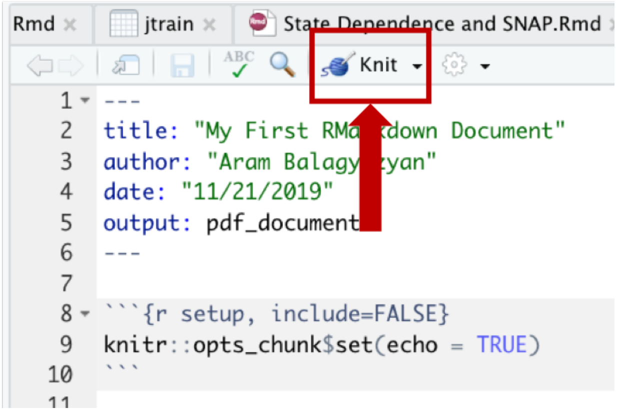
( ) PDF
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

( ) Word
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

- Upon clicking OK, RStudio will open up a new R markdown template document.

Using R Markdown to Combine Documents with Code and Code Output

- Save the markdown document somewhere in your hard drive and click on `Knit`.



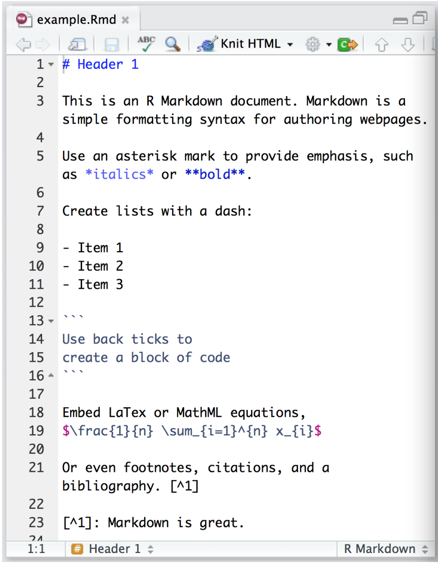
~~~
---
title: "My First RMarkdown Document"
author: "Aram Balagyozyan"
date: "11/21/2019"
output: pdf_document
---


```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
~~~

Using R Markdown to Combine Documents with Code and Code Output

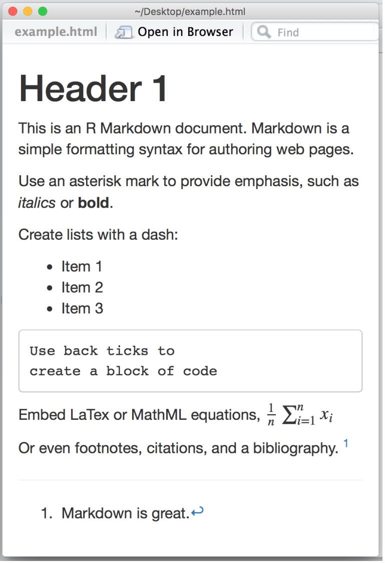
- As a result, R will create an RMD file + few auxiliary files and store them under the directory where the R markdown document is located.
- Among those auxiliary files is the output document. In this case, it's an HTML file with the same name as the markdown file.

R Markdown Basics

- For example, the file on the left shows basic R Markdown and the resulting output on the right:



```
# Header 1

This is an R Markdown document. Markdown is a
simple formatting syntax for authoring webpages.

Use an asterisk mark to provide emphasis, such
as *italics* or **bold**.

Create lists with a dash:

- Item 1
- Item 2
- Item 3

```
Use back ticks to
create a block of code
```

Embed LaTex or MathML equations,
$\frac{1}{n} \sum_{i=1}^{n} x_{i}$

Or even footnotes, citations, and a
bibliography. [^1]

[^1]: Markdown is great.
```



Header 1

This is an R Markdown document. Markdown is a simple formatting syntax for authoring web pages.

Use an asterisk mark to provide emphasis, such as *italics* or **bold**.

Create lists with a dash:

- Item 1
- Item 2
- Item 3

```
Use back ticks to
create a block of code
```

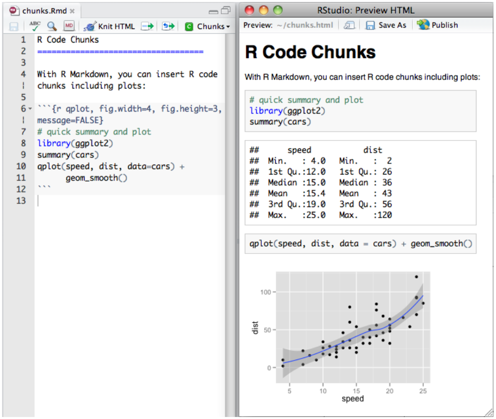
Embed LaTex or MathML equations, $\frac{1}{n} \sum_{i=1}^{n} x_i$

Or even footnotes, citations, and a bibliography. [1]

1. Markdown is great.↩

R Code Chunks

- Within an R Markdown file, R Code Chunks can be embedded with the native Markdown syntax for fenced code regions. For example, the following code chunk computes a data summary and renders a plot as a PNG image:



~~~
R Code Chunks
=================================

With R Markdown, you can insert R code
chunks including plots:

```{r qplot, fig.width=4, fig.height=3,
message=FALSE}
# quick summary and plot
library(ggplot2)
summary(cars)
qplot(speed, dist, data=cars) +
      geom_smooth()
```
~~~

R Code Chunks

With R Markdown, you can insert R code chunks including plots:

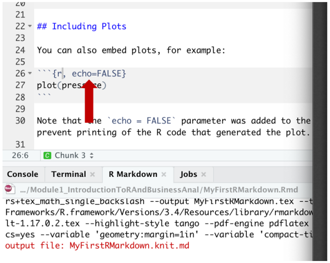
```
# quick summary and plot
library(ggplot2)
summary(cars)
```

```
##      speed          dist
## Min.   : 4.0   Min.   :  2
## 1st Qu.:12.0   1st Qu.: 26
## Median :15.0   Median : 36
## Mean   :15.4   Mean   : 43
## 3rd Qu.:19.0   3rd Qu.: 56
## Max.   :25.0   Max.   :120
```

```
qplot(speed, dist, data = cars) + geom_smooth()
```

R Code Options

- Sometimes, you may need to execute some R code inside an R Markdown document but you want to hide the code and display only the output.
- The `echo=FALSE` option does the trick

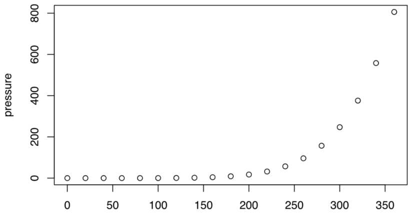


~~~
## Including Plots

You can also embed plots, for example:

```{r, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the
prevent printing of the R code that generated the plot.
~~~

```
…/Module1_IntroductionToRAndBusinessAnal/MyFirstRMarkdown.Rmd
rs+tex_math_single_backslash --output MyFirstRMarkdown.tex --t
Frameworks/R.framework/Versions/3.4/Resources/library/rmarkdow
lt-1.17.0.2.tex --highlight-style tango --pdf-engine pdflatex
cs=yes --variable 'geometry:margin=1in' --variable 'compact-ti
output file: MyFirstRMarkdown.knit.md
```

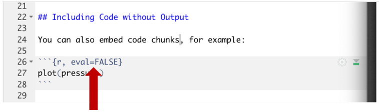
Including Plots

You can also embed plots, for example:

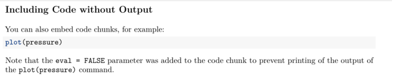


R Code Options

- Some other times, you may need to simply display some R code inside an R Markdown document without displaying the output.
- The `eval=FALSE` option does the trick



~~~
## Including Code without Output

You can also embed code chunks, for example:

```{r, eval=FALSE}
plot(pressure)
```
~~~



Including Code without Output

You can also embed code chunks, for example:

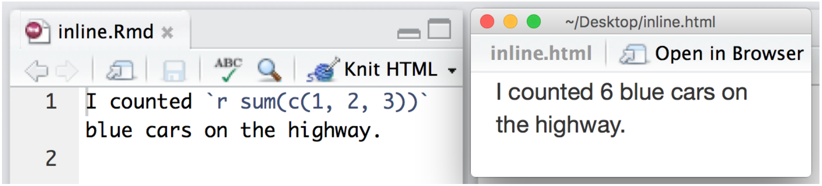
```
plot(pressure)
```

Note that the `eval = FALSE` parameter was added to the code chunk to prevent printing of the output of the `plot(pressure)` command.

- If neither option is specified, then RStudio will produce both the code and output.

Inline R Code

- You can also evaluate R expressions inline by enclosing the expression within a single back-tick qualified with `'r'`. For example, the following code embeds R results as text in the output at right

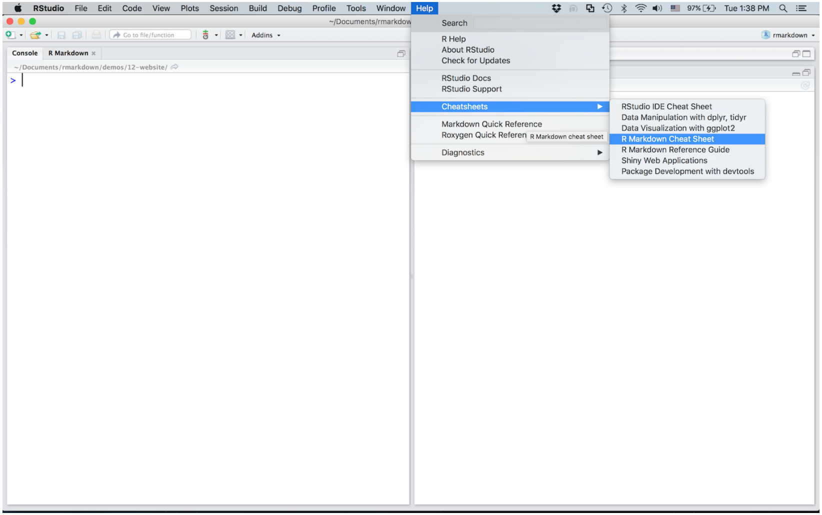


```
I counted `r sum(c(1, 2, 3))`
blue cars on the highway.
```

I counted 6 blue cars on the highway.

Cheat Sheets

- For a quick reference on R Markdown and other features of RStudio, Go to *File* > *Help* > *Cheatsheets* > *R Markdown Cheat Sheet* to open the main R Markdown cheatsheet.



Outline

Getting Help

- R has an integrated help system that you can use to know more about the system and its functionality
- You can issue help.start() at the prompt to launch a browser showing the HTML help pages.
- Another way of getting help is to use the help() function.
- For instance, if you want some help on the plot() function you can type the following command at the prompt

```
help(plot)
# or alternatively
?plot
```

- Note that when you type a # inside any code, R treats everything on the line that follows # as a comment and does not attempt to evaluate/execute it.
- A powerful alternative is to use the RSiteSearch() function that searches for key words or phrases in the mailing lists, R manuals, and help pages. For example, try typing

```
RSiteSearch('ggplot2')
```

Getting Help and Common Problems.

- For more direct question related to R, stackoverflow is a must.