Live Session 1

Introduction to R

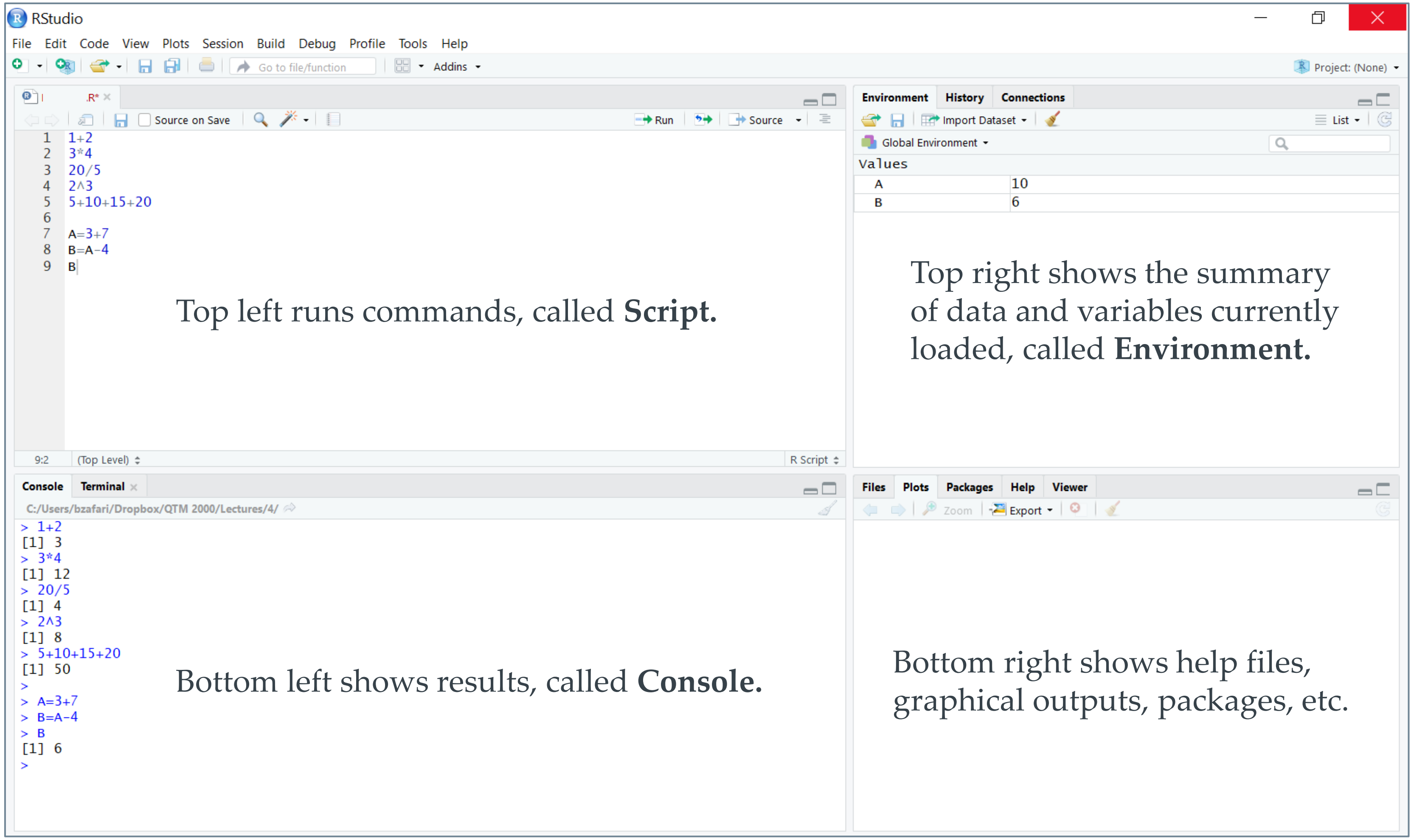
Babak Zafari
August 29, 2024

GEORGETOWN UNIVERSITY / McDONOUGH SCHOOL of BUSINESS

Outline

- Goal 1: Get set up in R and RStudio
- Goal 2: Load data
- Goal 3: Identify three major data types
- Goal 4: Identify four data structures
- Goal 5: Describe data

Goal 1: Get set up in R and RStudio



```
1+2
3*4
20/5
2^3
5+10+15+20

A=3+7
B=A-4
B
```

Top left runs commands, called **Script.**

Values

A	10
B	6

Top right shows the summary of data and variables currently loaded, called **Environment.**

C:/Users/bzafari/Dropbox/QTM 2000/Lectures/4/

```
> 1+2
[1] 3
> 3*4
[1] 12
> 20/5
[1] 4
> 2^3
[1] 8
> 5+10+15+20
[1] 50
>
> A=3+7
> B=A-4
> B
[1] 6
>
```

Bottom left shows results, called **Console.**

Bottom right shows help files, graphical outputs, packages, etc.

Goal 2: Loading the Data

- Refer to the last slide before the exercise.

Goal 3: Identify three major data types

- We can use the following functions to find about data types:
  - `str( )`
  - `class( )`

Goal 4: Identify four data structures

- Functions and methods that were reviewed in this goal:
  - `length( )`
  - `is.na( )`
  - `data.frame( )`
  - `names( )`

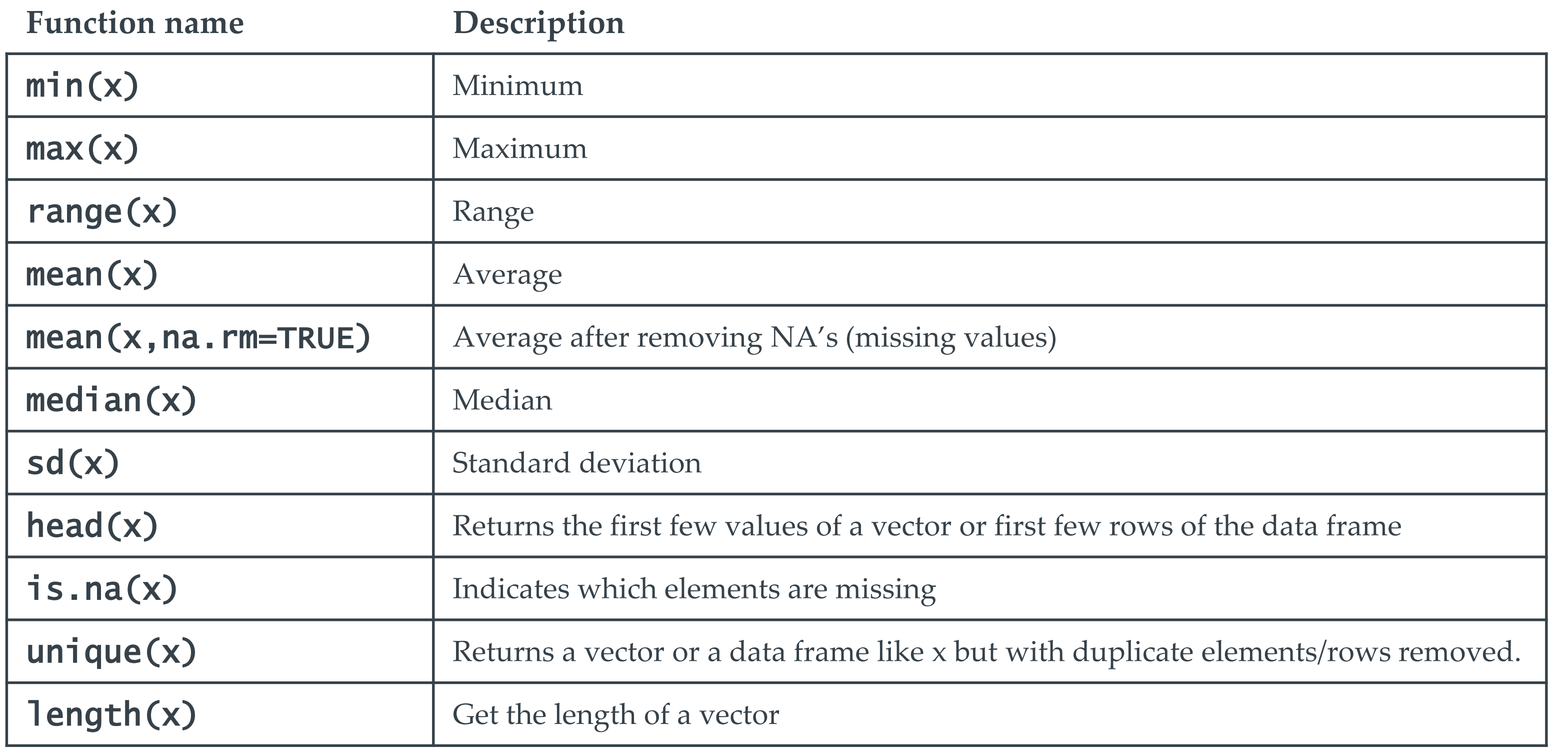
  - **Use of [ ] versus $ to identify subsets (i.e., variables)**

Goal 5: Describe data

- Functions and methods that were reviewed in this goal:
  - `summary( )`
  - `mean( )`
  - `mean( ,na.rm=TRUE)`
  - `median( )`
  - `min( )`
  - `max( )`
  - `unique( )`
  - `table( )`

  - `apply(x,MARGIN,FUN)`
    - Takes a data frame or a matrix as an input and gives output in vector (list or array). Takes three arguments:
      - x: a matrix, an array, or a data frame
      - MARGIN: if 1, the operation is performed on rows, and if 2, the operation is performed on columns
      - FUN: the function to apply. Built-in functions like mean, sum, etc. and user-defined functions

Data Summary Functions



Function name	Description
`min(x)`	Minimum
`max(x)`	Maximum
`range(x)`	Range
`mean(x)`	Average
`mean(x,na.rm=TRUE)`	Average after removing NA's (missing values)
`median(x)`	Median
`sd(x)`	Standard deviation
`head(x)`	Returns the first few values of a vector or first few rows of the data frame
`is.na(x)`	Indicates which elements are missing
`unique(x)`	Returns a vector or a data frame like x but with duplicate elements/rows removed.
`length(x)`	Get the length of a vector

Warnings in R

- What does + mean in console?
  - You probably forgot to add a “)” at the end of a function or ran a partially-highlighted function. Hit Esc and re-run the line.

- “Unknown or uninitialised column: `category`”
  - Either the data is not loaded or you’re using a incorrect name in referring to the data and/or its variables.

- Check this page for the list of most common warnings and errors in R and instructions on how to deal with them: LINK

Loading the Data (IMPORTANT)

- In importing datasets to R, you can take different approaches.

Either do this:

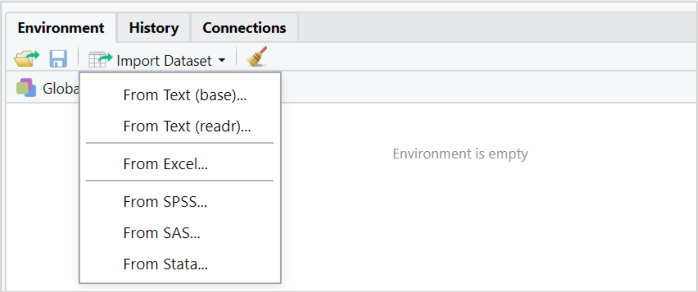
1. Save your R script in a folder.
2. Copy/paste the dataset to the folder where the R script is stored.
3. In RStudio, go to “Session → Set Working Directory → To Source File Location“
4. Run this code if the file has .csv format: **WV = read.csv("WorldValues_W1Live.csv")**

Or:

1. Run this code by replacing the address to the local address where the .csv file is stored:
2. **WV = read.csv("C:/…/ WorldValues_W1Live.csv")**

Or:

- Use the Import Dataset option from the top right window.
- For .csv file choose either of the “From Text” options.
- For .xls or .xlsx files choose “From Excel” option.
- Read more about these options here: [LINK]



Exercise

Load and save ***WorldValues_W1Live.csv*** as "WV".

1. Does this dataset have missing values? If so, how many missing values does it have?
2. Create an object that contains unique birth years. How many unique birth years the dataset have?
3. Create a new variable that stores the age of each respondent at the time of the survey. (add/attach this variable to the dataset)
4. What is the average age of survey respondents?
5. How many responses are from country 12? The variable 'Code' captures country code for responses.
6. What is the average age of respondents from country 12?

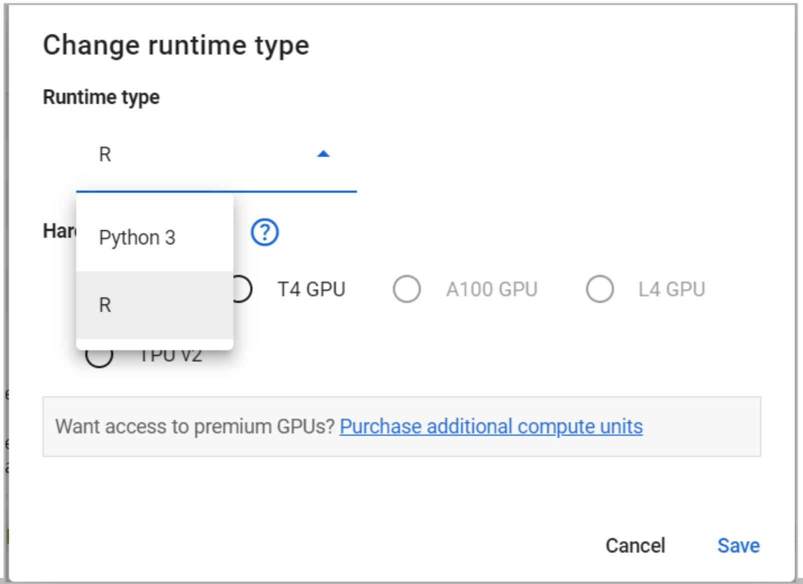
7. Final step (OPTIONAL):

   Once you're done with these steps, move the working codes to the markdown file named ***Markdown_W1Live.rmd***. Put the code of question in its code chunk in the markdown document. Knit this document to HTML. Open the HTML file in a browser and save/print that page as a PDF file.

How to Run R Code in Google Colab

- **What is Google Colab?**
  - A free, cloud-based platform provided by Google that allows you to write and execute code in an interactive environment.
  - It supports multiple programming languages, including Python and R, and is ideal for coding, data analysis, and machine learning.

- **Setting up R in Google Colab:**
  - **Step 1: Open a New Notebook:** Go to Google Colab and click on "New Notebook."
  - **Step 2: Set the Runtime to R:** By default, Google Colab runs Python code. To run R code, go to Runtime → Change runtime type → Choose R from the list of Runtime types



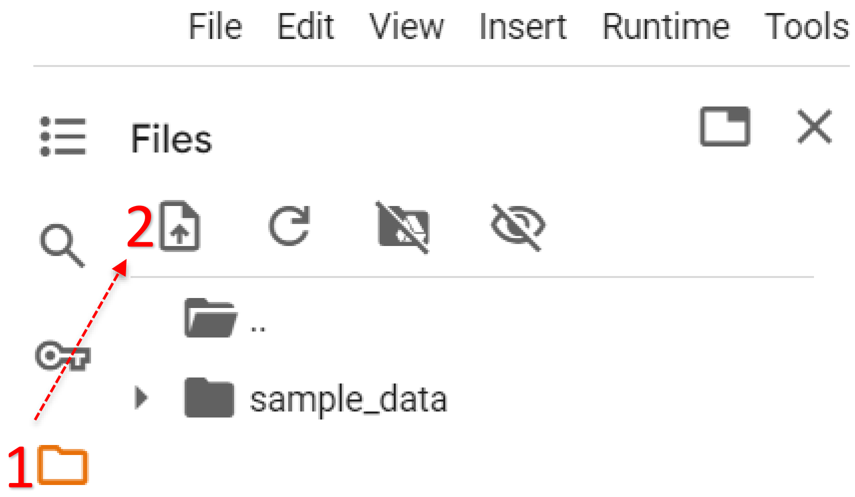
Change runtime type

Runtime type: R

Want access to premium GPUs? Purchase additional compute units

How to Load Data

- **Step 1**: In your Colab notebook, click on the Folder icon (shown in orange in the screenshot)
- **Step 2**: Click the Upload button to browse and upload your data file from your local computer.
  - Once uploaded, the file is stored temporarily in the Colab environment. You can load the data into your R environment using standard R commands.
  - Files uploaded to Colab are stored temporarily. If you disconnect or restart the notebook, the uploaded files will be lost. You'll need to re-upload the data file every time you start or reconnect to the notebook.



How to Print Your Output (for assignment submission)

- Step 1: Organize Your Code:
  - Copy your R code from your development environment (e.g., RStudio) and paste it into the appropriate cells in your Google Colab notebook. Ensure each logical block of code is placed in its own cell for clarity and organization.
- Step 2: Run All Code Cells
  - Before submitting, it's essential to run all your code cells to ensure everything works as expected and that all outputs are generated.
  - To Run All Cells Go to the top menu and click on Runtime. Select Run all from the dropdown menu. This will execute all the cells in your notebook from top to bottom.
  - *Important: Make sure you have uploaded your data files (or connected to Google Drive) before running the cells that require data.*
- Step 3: Review Outputs
  - After running all cells, scroll through the notebook to ensure all outputs (e.g., tables, plots, summaries) are correctly generated.If any errors occur, resolve them and re-run the affected cells to confirm that everything works properly.
- Step 4: Convert to PDF for Submission
  - Once all code has been successfully run and outputs are generated, you need to *print the notebook as a PDF*. In the Colab menu, go to File > Print. In the print dialog, choose Save as PDF as the destination. Save the PDF file to your computer.
  - Submit Your PDF: Upload the PDF file along with any other required files (e.g., .R scripts) to your assignment submission portal.

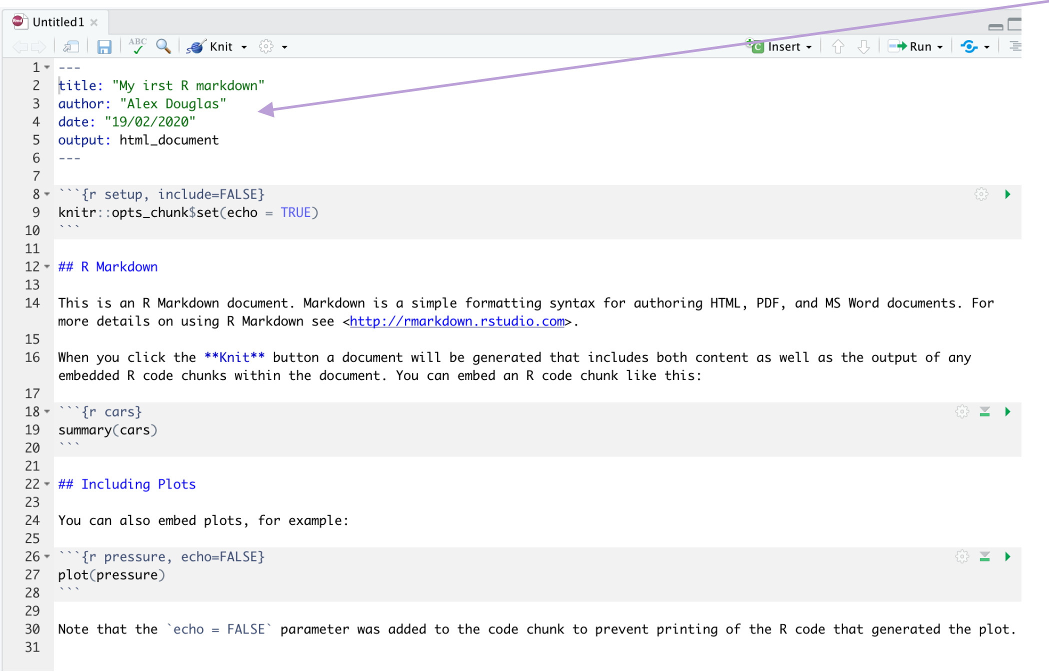
The next few slides are OPTIONAL and will teach you about R Markdown and how to create them.

https://intro2r.com/new_rm.html

Creating a R Markdown Document (OPTIONAL)

- Within RStudio, click on the menu File → New File → R Markdown…. In the pop up window, give the document a ‘Title’ and enter the ‘Author’ information (your name) and select HTML as the default output. We can change output type later.

- When a new R markdown document is created it includes some example R markdown code. You can highlight and delete everything in the document except the information at the top between the --- delimiters, also called the YAML header.



~~~
---
title: "My irst R markdown"
author: "Alex Douglas"
date: "19/02/2020"
output: html_document
---

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

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For
more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any
embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

Including Plots

You can also embed plots, for example:

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

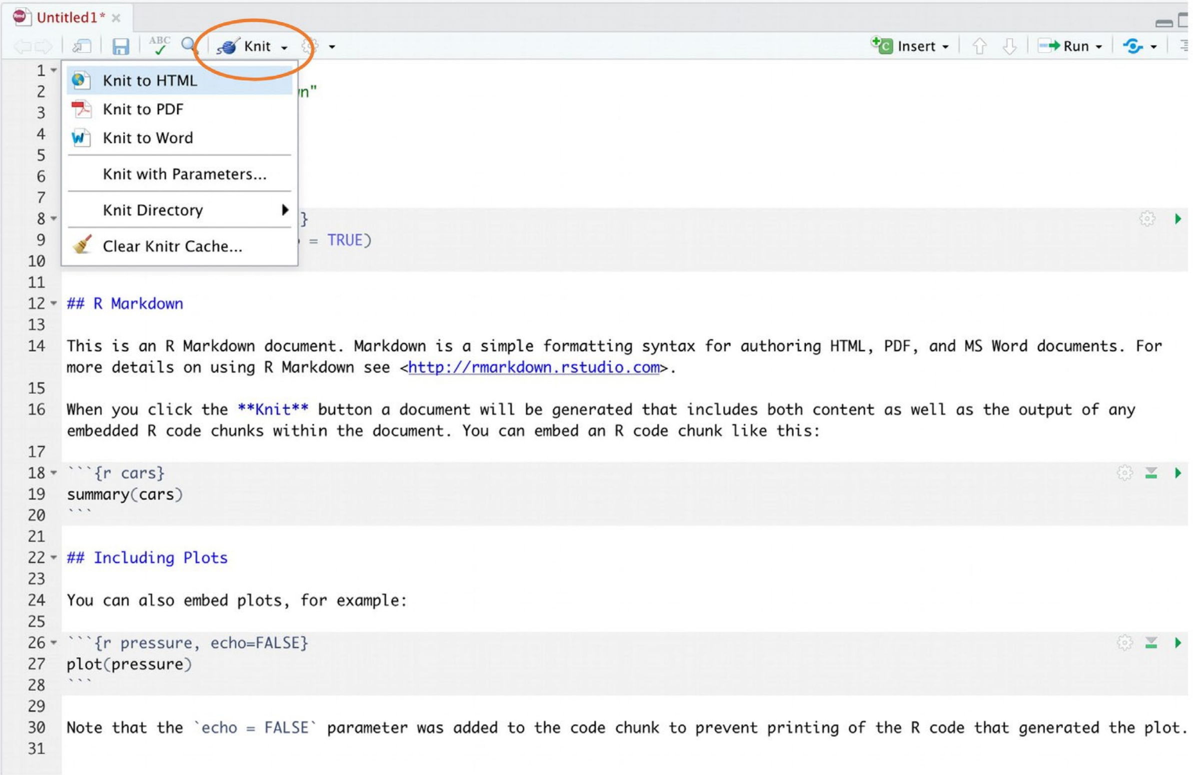
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```

https://intro2r.com/new_rm.html
```

Creating a R Markdown Document (OPTIONAL)

- Once you have created and saved your R markdown document (a file with .Rmd extension), you can convert the file to a HTML or PDF document by clicking at the black triangle next to the Knit icon at the top of the source window and select the appropriate option:



~~~
n"





}
= TRUE)


## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For
more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any
embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
~~~

Including Plots

You can also embed plots, for example:

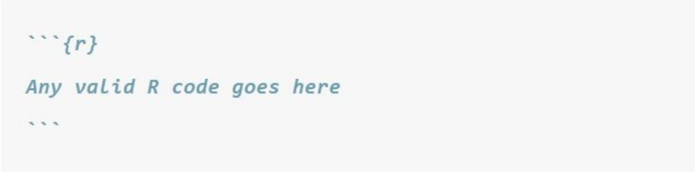
```{r pressure, echo=FALSE}
plot(pressure)
```

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

https://intro2r.com/new_rm.html
```

R Markdown Code Chunks (OPTIONAL)

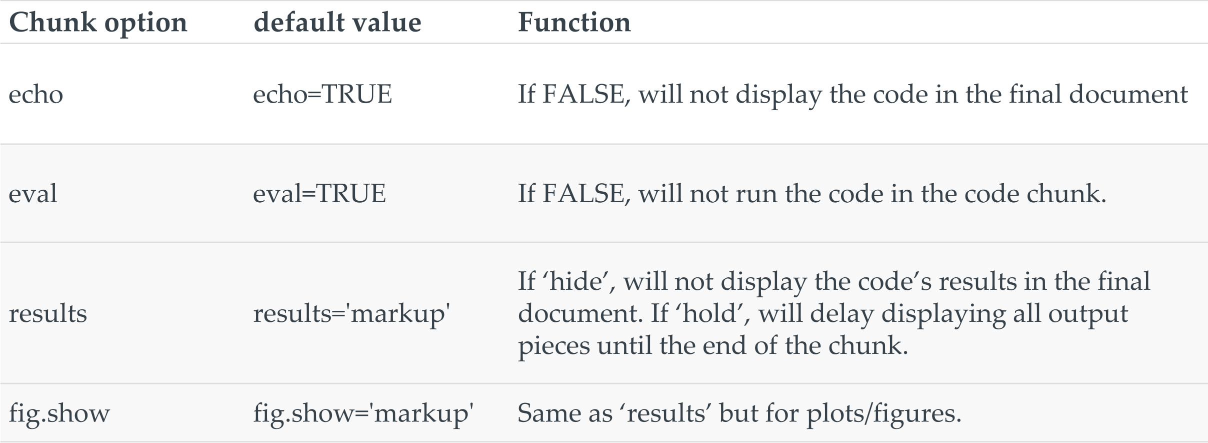
- To include R code into your R markdown document you simply place your code into a ‘code chunk’. All code chunks start and end with three backticks ```.



~~~
```{r}
Any valid R code goes here
```
~~~

- You can insert a code chunk by either typing the chunk delimiters ```{r} and ``` manually or use the RStudio toolbar (the Insert button) or by clicking on the menu Code → Insert Chunk

- Some of the code chunk options: →
  - These options could be added in front of ‘r’ in the beginning of the code chunk, separated by a comma.



Chunk option	default value	Function
echo	echo=TRUE	If FALSE, will not display the code in the final document
eval	eval=TRUE	If FALSE, will not run the code in the code chunk.
results	results='markup'	If ‘hide’, will not display the code’s results in the final document. If ‘hold’, will delay displaying all output pieces until the end of the chunk.
fig.show	fig.show='markup'	Same as ‘results’ but for plots/figures.

https://intro2r.com/new_rm.html

R Markdown files are stand-alone (OPTIONAL)

- Every R Markdown file (Rmd file) must be completely stand-alone. It doesn't share any information with the Console or the Environment that you see in your RStudio session. All R code that you need to do whatever you are trying to do must be included in the Rmd file itself.

- For example, if you use the point-and-click user interface in the RStudio Environment tab to import a data file, the read-in dataset will not be available for use within your Rmd file.

- Similarly, if you load a package by typing "library(mosaic)" (or "require(mosaic)") in the Console window, mosaic functions and data will not be available to use within the Rmd file.

- NOTE: By default, the working directory for your R Markdown document is the directory in which that your markdown file is stored.