

# **Week-1: Introduction**

**NM2207: Computational Media Literacy**

**Sanka Rasnayaka**

**Lecturer, School of Computing, NUS**



**NUS**  
National University  
of Singapore

Faculty of Arts  
& Social Sciences

# **Week-1: Introduction**

**NM2207: Computational Media Literacy**

**Narayani Vedam, Ph.D.**

**Department of Communications and New Media**



**Faculty of Arts  
& Social Sciences**



# This week

# This week

**I. Introduction to people (click here)**

**II. Introduction to the course (click here)**

**III. Preliminaries (click here)**

**IV. Computational thinking (click here)**

**V. R Studio (click here)**

**VI. R Markdown (click here)**

# I. Meet the people

# About NV

I am Narayani Vedam, you can address me by my initials, NV, for brevity's sake 😊 !

You can learn more about me and my work [here](#)

## Academic Background

1. Postdoctoral Fellow, Department of Communications and New Media
2. Ph.D., Department of Aerospace Engineering, The Indian Institute of Science, Bengaluru
3. M.S., Department of Electrical and Computer Science, Texas A&M University, College Station

## Research Interests

1. Human behaviour on online social platforms
2. Multi-agent systems and controls

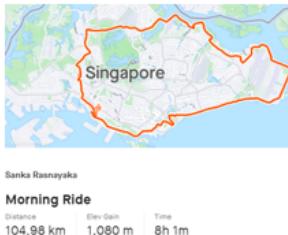
# About Sanka

## Sanka Rasnayaka

- PhD from SoC, NUS, 2021
- Biometric Authentication, AI & Computer Vision
- TA/Lecturer: 1101, 1231, 1010, 2040, 3230, 5332
- Music, Drawing, Running/Cycling, Reading Fiction

<https://www.comp.nus.edu.sg/~sanka/>

sanka@nus.edu.sg



# Know your Tutors

## Naman Agrawal

Hello!

- I am a third-year undergraduate student at the National University of Singapore.
- I major in **Data Science and Analytics** and **Economics** with a minor in Computer Science
- I enjoy experimenting with audio signal processing, 3D computer vision, and econometric modelling to develop novel insights.
- I'm currently working on developing complex-valued convolutional neural networks for applications in musical engineering.



# Know your Tutors

**TA-2**

**TBD**

- ...



## II. All about the course

# About the course: background

Course is for **non-programmers**

Literacy through **hands-on activity**

Learning to **code** in 

## Why learn coding?

Many real-world jobs rely on programming;

- to engage with large-scale data
- to complete statistical analyses

# About the course: background

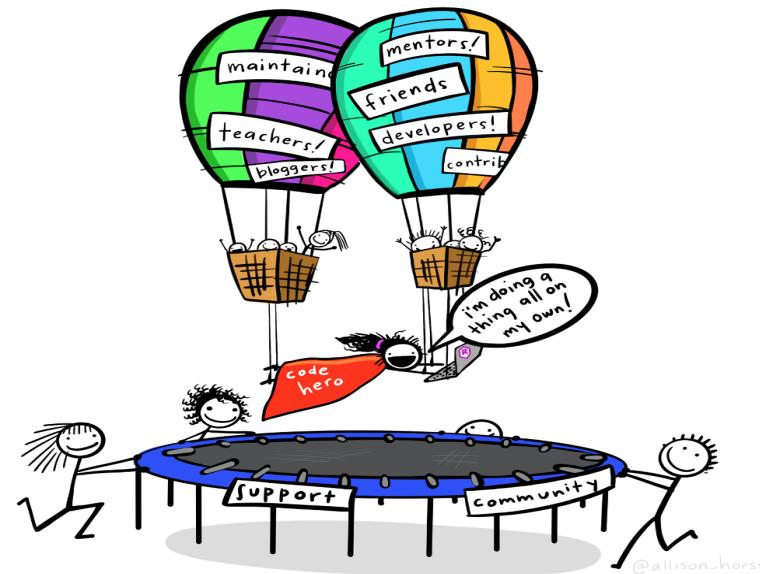
Course is for **non-programmers**

Literacy through **hands-on activity**

Learning to **code** in 

## Why ?

- Great data manipulation and visualization suite
- Comprehensive statistical packages
- Has a vibrant online community
- **Easiest** language for beginners
- Open source



Source: <https://allisonhorst.com/everything-else>

# About the course: contents

## Part-1: Data

- **Week-1:** Introduction to NM2207
- **Week-2:** Introduction to data & visualization
- **Week-3:** Variables & its types
- **Week-4:** Manipulating data
- **Week-5:** Functions
  - **Alert:** Introduction to github (not graded)
- **Week-6:** Iterations
  - **Alert:** Creating a web portfolio (not graded)

## Part-2: Visualization

- **Week-7:** Visualization with `ggplot2`
  - **Alert:** Work on project proposal!!
- **Week-8:** Visualization with `shiny`
- **Week-9:** Exploratory data analyses
  - **Submission alert:** Project proposal!!
- **Week-10 onward:** Practice project(s)

# About the course: components

4-MC, Workload: approx. 10hrs weekly

## 1. Video "lectures"

- Explanatory videos of various concepts
- Code along with videos (templates will be provided)
- Videos to be watched before attending classes
- **Average time:** 4hrs

# About the course: components

4-MC, Workload: approx. 10hrs weekly

## 2. Classroom "challenges"

- Recommend to work in **GROUPS**
  - Group sizes can be 2/3
- Questions can be,
  - Coding, narrations, MCQs
- Tutors will be present to guide
- Solution Templates in R Markdown for submissions will be provided
- **Average time:** 3hrs

## 1. Video "lectures"

# About the course: components

4-MC, Workload: approx. 10hrs weekly

## 3. Final Project

A data story implementing concepts learned in class + new concepts you self-learn

a. It could be a webpage;

- data visualization conveying a story
- Example: <https://hedonometer.org/>

b. It could be a game;

- an app that lets the users interact, score points, chat, etc.
- Example: <https://shiny.posit.co/r/gallery/miscellaneous/hex-memory/>

## 1. Video "lectures"

## 2. Classroom "challenges"

# An instance of a data story

1. Ask a question
2. Use data to answer

RESEARCH ARTICLE | PSYCHOLOGICAL AND COGNITIVE SCIENCES | 

**f     **

**The emotional and mental health impact of the murder of George Floyd on the US population**

Johannes C. Eichstaedt , Garrick T. Sherman , Salvatore Giorgi, , and Sharath Chandra Guntuku  

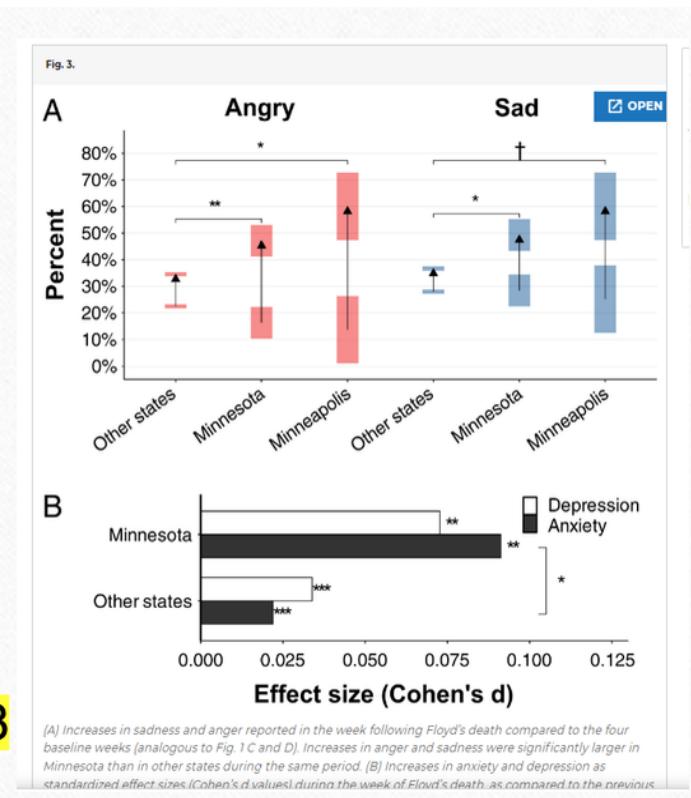
[Authors Info & Affiliations](#)

Edited by Douglas S. Massey, Princeton University, Princeton, NJ, and approved July 21, 2021 (received for review June 3, 2021)

September 20, 2021 | 118 (39) e2109139118 | <https://doi.org/10.1073/pnas.2109139118>

**THIS ARTICLE HAS BEEN UPDATED**

<https://www.pnas.org/doi/10.1073/pnas.2109139118>



# About the course: submission policy

## 1. Challenges

- Complete Week N's challenge by the end of Week N's tutorial;
  - Inform the TA and submit by,
    - the end of class for 100% grades
    - class-day midnight for 50% grades

# About the course: submission policy

## 2. Weekly assignments (if any)

- On time submissions, 100% grades
- Late submissions upon notifying TA, 50% grades

## 1. Challenges (Attendance/Participation)

# About the course: submission policy

## 3. Quiz

- Only one quiz during **Week 6**

## 1. Challenges (Attendance/Participation)

## 2. Weekly assignments (if any)

# About the course: submission policy

## 4. Final Project

- **Week 9:** Intermediate submission
  - Proposal with data sources and plan for data preparation
- **Week 10 onward:** Progress report
  - Write-up on the progress of the project
  - Highlight *show-stoppers*
- **Week 13:** Final project
  - A working website conveying the story
  - Documentation of the idea and execution

## 1. Challenges (Attendance/Participation)

## 2. Weekly assignments (if any)

## 3. Quiz

All project submissions are expected on friday of the respective week by 17:00hrs

# About the course: evaluation

## 1. Tutorial/Participation: 30%

- attendance,
- preparation,
- and other evidence

# About the course: evaluation

## 2. Quiz: 30%

- only one quiz
- will be in week-6, includes syllabus of weeks 1-6

## 1. Tutorial/Participation: 30%

# About the course: evaluation

## 3. Project: 40%

- Final write-up: 25%
  - Background/Approach/Final outcome
- Demonstrated understanding: 20%
- Complexity: 15%
- Demonstrated learning: 15%
- Diary entries (weeks 9, 10, 11, 12): 10%
- Creativity: 10%
- Works well: 5%

## 1. Tutorial/Participation: 30%

## 2. Quiz: 30%

# How to ace?

Like with everything else in life, practice helps!

- Practice regularly
- Use Google and the online resources extensively
- Work with a friend(s)
- Do not hesitate to **ASK** questions
  - If you are shy to do so in person, use Canvas
  - Every week a new discussion thread will be initiated
  - Feel free to post questions, however insignificant you may think!
  - Please check for repetitions before posting

# Learning outcomes

a. Mine insights from data-sets of interest

- Involves the ability to curate, manipulate, and analyze data using 

b. Visualize useful insights through plots and dashboards

c. Communicate the insights more comprehensively through an interactive website using R Markdown and Shiny

- Without the need to learn core HTML5 technologies

# III. Preliminaries

# Files

- They are objects on the computer
- They store data or commands for a computer program
  - They could be an application/data/system file
  - Extensions are used to recognise the file type
  - Examples: .docx, .xlsx, .txt, .jpeg, .png, .csv, etc.
- They are created using a computer software program
  - Examples:
    - To create a text file, we use a text editor
    - To create an image file, we use an image editor
    - To create a document, we use a document processor
- They are stored in the memory of the computer

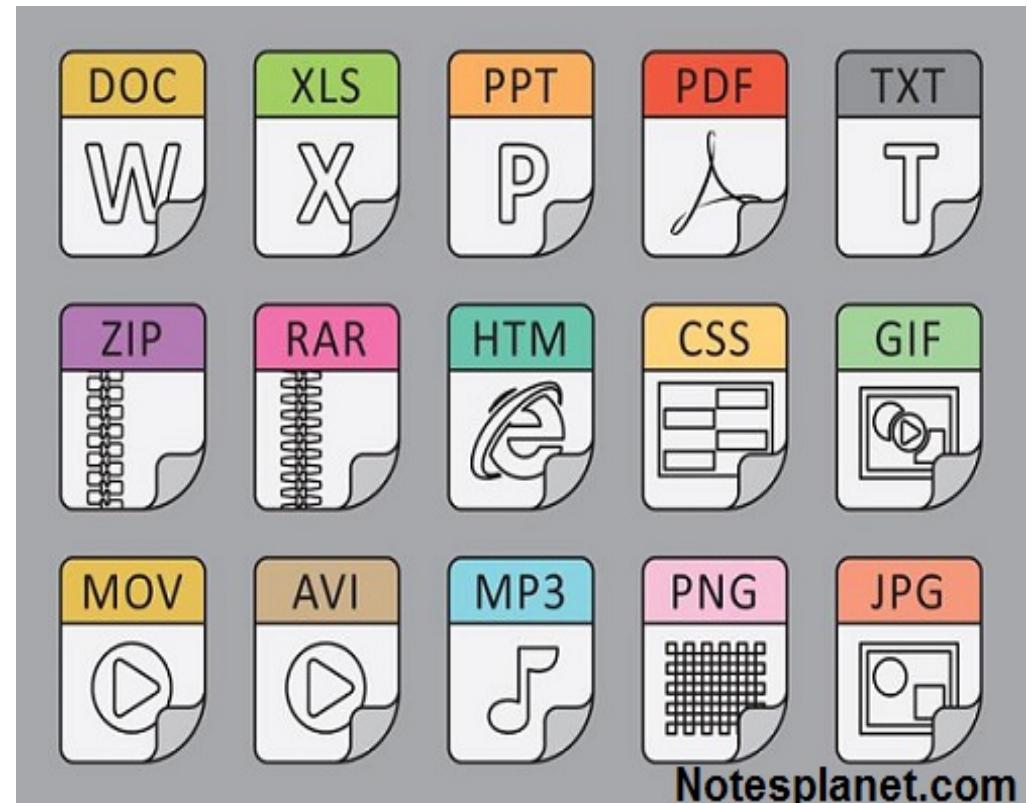


Figure: Different types of files and their extensions

# Folders

- Folders can contain file(s) or other folders
- They can be thought of as storage boxes/organisers

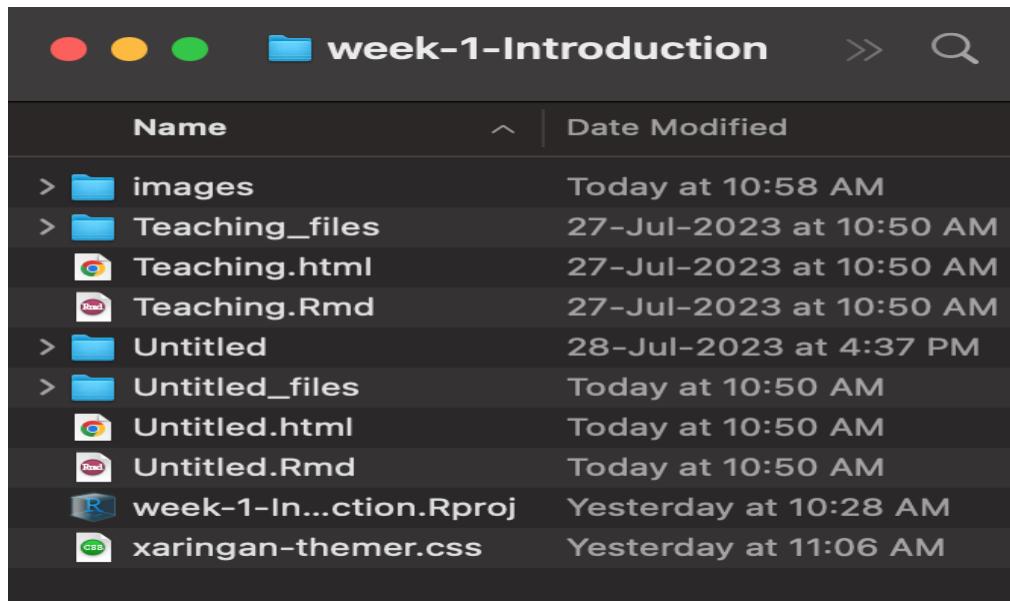


Figure: A folder in macOS

	Name	Date modified	Type	Size
s	random	9/9/2022 4:20 pm	File folder	
s	selected-downloads	16/10/2022 3:25 pm	File folder	
s	Soc	19/10/2022 8:49 am	File folder	
s	ss	7/11/2022 9:20 pm	File folder	
s	stuff	5/7/2021 11:23 am	File folder	
s	test	1/3/2022 5:03 pm	File folder	
s	test01	7/5/2022 1:19 am	File folder	
s	Testing	30/7/2022 1:47 pm	File folder	
s	Zoom BG	23/9/2021 12:15 am	File folder	
ArcGIS Pro	2/9/2022 6:40 pm	Shortcut	2 KB	
bitlocker.pdf	25/8/2021 10:50 am	Microsoft Edge PD...	176 KB	
bown transcript.docx	17/10/2022 4:13 pm	Microsoft Word D...	53 KB	
CNMSOC HOTO.docx	2/9/2021 1:19 pm	Microsoft Word D...	15 KB	
Core Temp	2/9/2022 6:40 pm	Shortcut	2 KB	
degree.docx	19/8/2021 7:14 pm	Microsoft Word D...	21 KB	
Discord	15/12/2022 11:05 ...	Shortcut	3 KB	
doge.gif	1/4/2022 3:23 pm	GIF File	5,800 KB	
Fallout Shelter	26/7/2022 9:49 pm	Internet Shortcut	1 KB	
gooddino.jpg	30/6/2022 9:47 am	JPG File	6 KB	

Figure: A folder in Windows OS

# Paths

- They are the addresses to different files or folders
- Examples
  - `C:/Users/nvedam` path to a folder named `nvedam` on a Windows machine
  - `/Users/nvedam` path to a folder named `nvedam` on a macbook
  - `/Users/nvedam/example.txt` path to a file named `example.txt` on a macbook
  - `https://www.w3schools.com/images/picture.jpg` it could also be a path to files on the internet

# IV. Computational thinking

# What does it entail?

Programmers use the computer to solve problems. This includes,

- a. Breaking down the problem into solvable steps
- b. Identifying the inputs and outputs
- c. Identifying the intermediate operations
- d. Writing down the steps identified (**Algorithm!**)
  - flowcharts
  - pseudocode

# Flowchart

Some guidelines for developing flowcharts,

- It can have only one start and stop symbol
- General flow is either,
  - Top to bottom, or
  - Left to right
- Arrows should not cross each other

Symbol	Symbol Name	Purpose
	Start/Stop	Used at the beginning and end of the algorithm to show start and end of the program.
	Process	Indicates processes like mathematical operations.
	Input/ Output	Used for denoting program inputs and outputs.
	Decision	Stands for decision statements in a program, where answer is usually Yes or No.
	Arrow	Shows relationships between different shapes.
	On-page Connector	Connects two or more parts of a flowchart, which are on the same page.
	Off-page Connector	Connects two parts of a flowchart which are spread over different pages.

Figure: Flowchart symbols and their meanings

# Flowchart: Example 1

An algorithm to add two numbers provided by the user and print the result;

- **Inputs:** num1, num2
- **Outputs:** sum
- **Operations:** addition, print

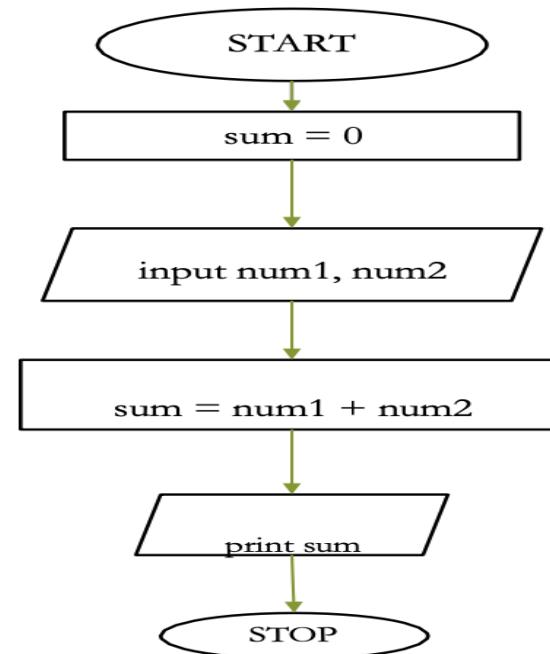


Figure: Flowchart for example 1

# Flowchart: Example 2

An algorithm to add ten numbers provided by the user and print the result;

- **Inputs:** num1, num2, ..., num10
- **Intermediary:** counter
- **Outputs:** sum
- **Operations:** addition, print

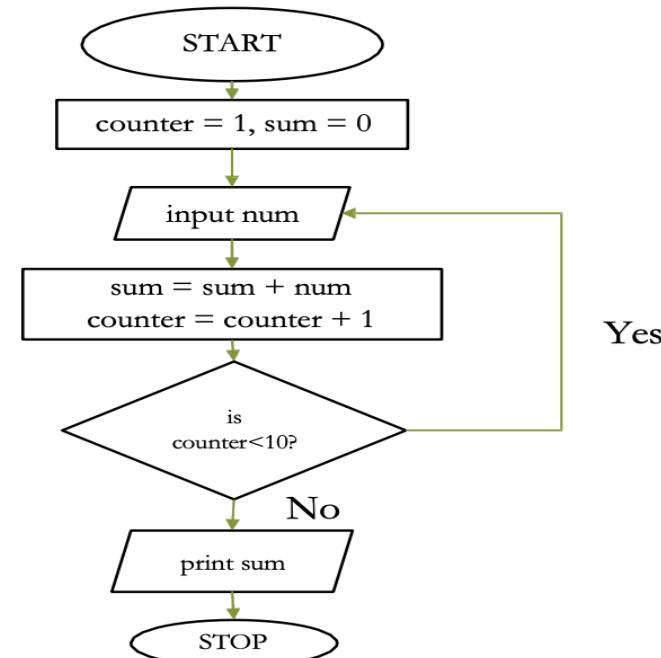


Figure: Flowchart for example 2

# Pseudocode

- The steps of a program
- The steps include the following components,
  - Variables (example: `num1`, `num2`, `sum`)
  - Assignment (example: `num1=0`, `sum=0`, `sum=a+5`)
  - Decision (example: `sum>0`)
  - Repetition (example: repeat some steps for more than 10 times)

# Pseudocode: Example 1

An algorithm to add two numbers provided by the user and print the result;

**Step-1:** Start

**Step-2:** Declare variables

```
sum=0
num1=0
num2=0
```

**Step-3:** Input values

```
print("enter values of num1 and num2 ")
```

**Step-4:** Perform the sum

```
sum = num1+num2
```

**Step-5:** Print the sum

```
print(sum)
```

**Step-6:** Stop

# Pseudocode: Example 2

Write pseudo code that tells a user that the number they entered is not a 5 or a 6

**Step-1:** Start

**Step-2:** Input num

**Step-3:**

```
if(num!=5 && num!=6)
{
    print("your number is not 5 or 6")
}
```

**Note:** && and != are called logical operators

**Step-3:** Stop

# Pseudocode: Example 3

Write pseudo code that tells a user that the number they entered is not a 5 or a 6

**Step-1:** Start

**Step-2:** Input num

**Step-3:**

```
if( num ==5 || num ==6)  
  
{go to **Step-5**}  
  
else  
  
{go to **Step-4**}
```

**Note:** || and == are called logical operators

**Step-4:** print("your number is not a 5 or a 6")

**Step-5:** Stop

# V. and R Studio

# What is R Studio

- It is an Integrated Development Environment, also called the IDE
- It adds a layer of aesthetics and functionality over 
- While installing them, always make sure that you,
  - Install 
  - Install R Studio

# Overview

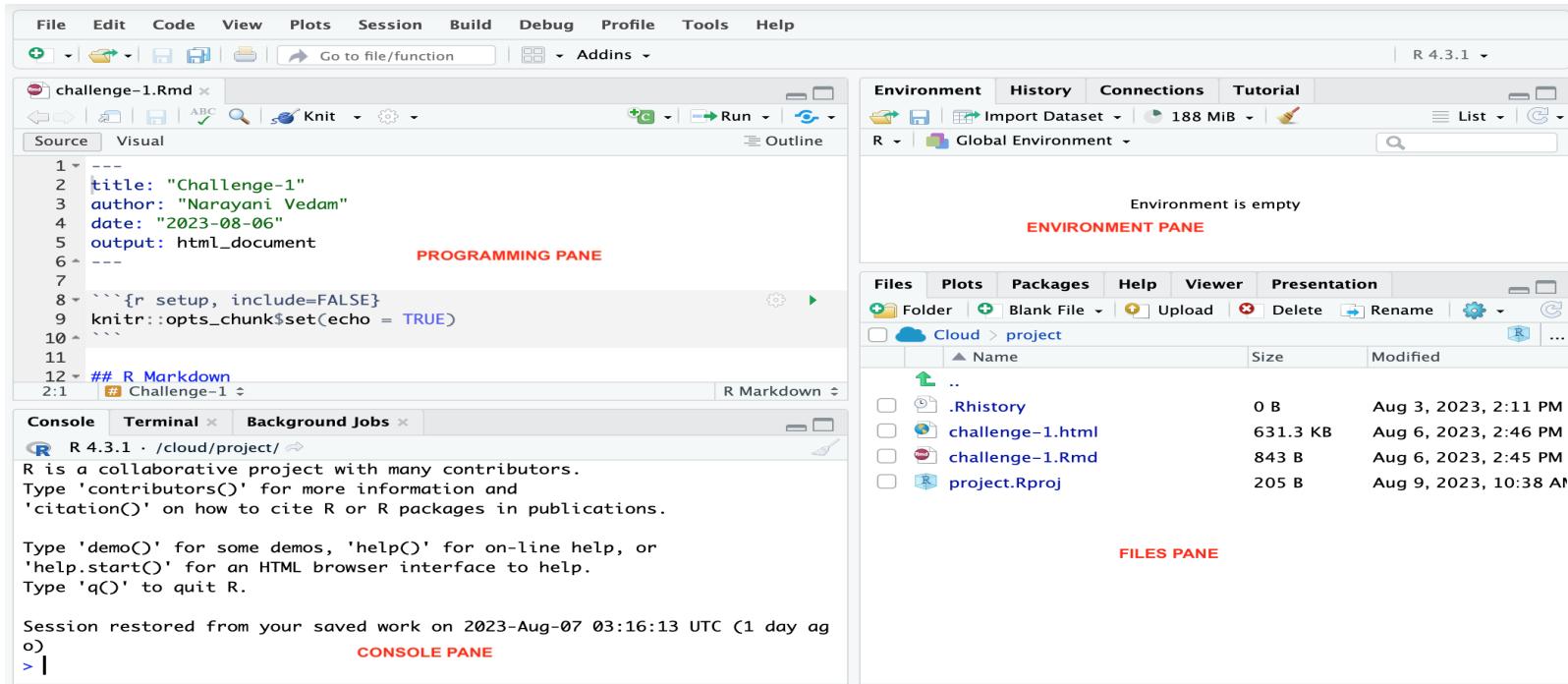


Figure: A typical layout of R Studio

# Overview

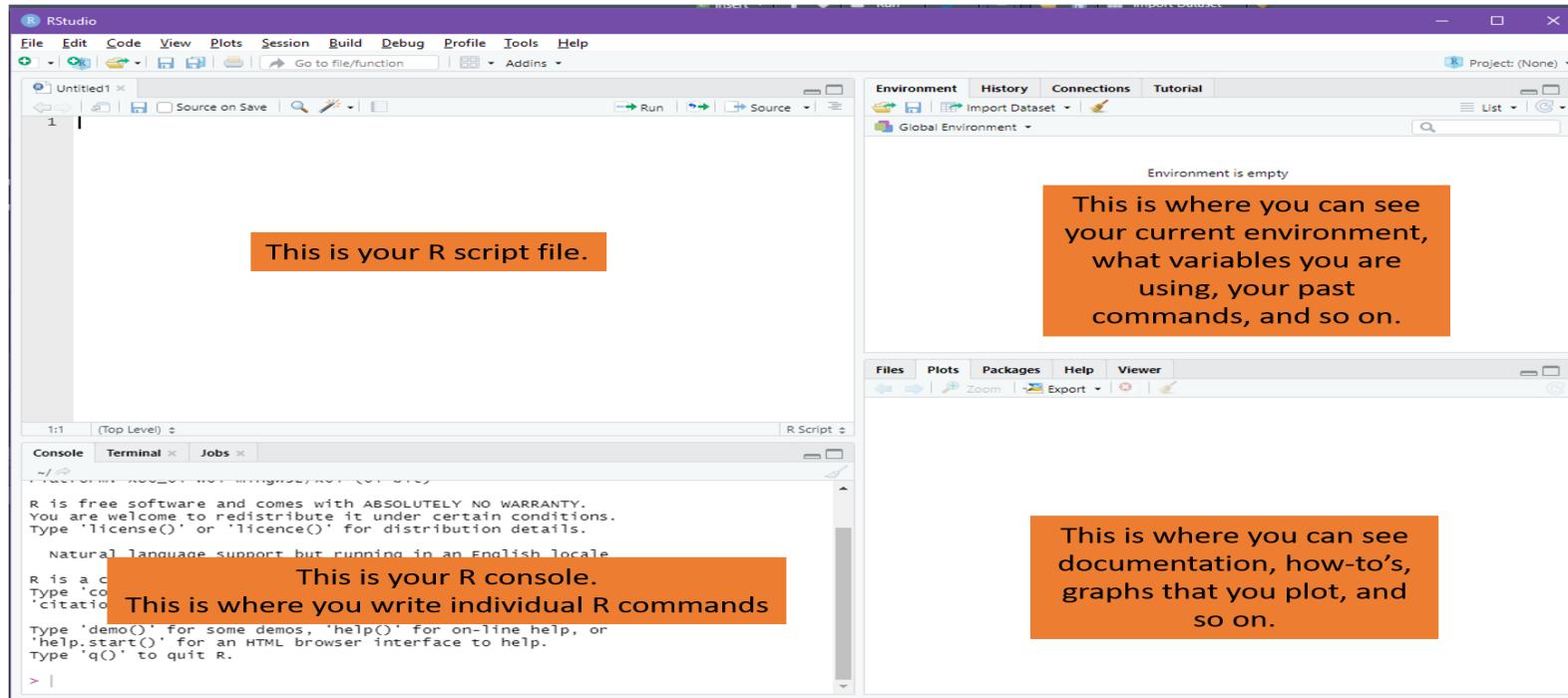


Figure: Different panes in the layout



## VI. R Markdown

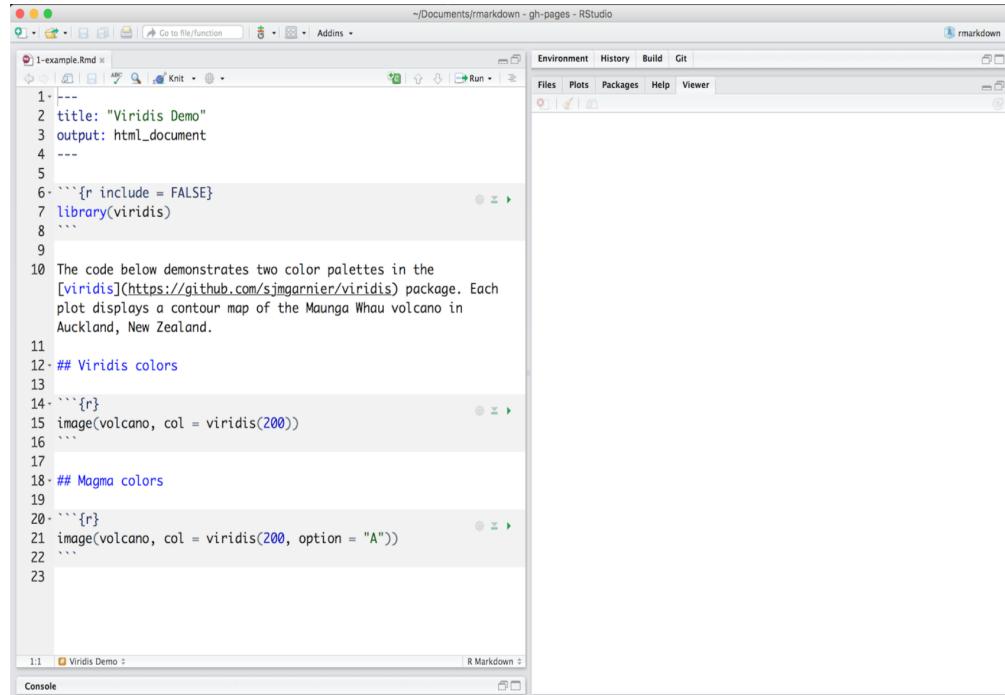
# Overview

- | It provides an authoring framework for data science.
- You can use a single R Markdown file to do the following
  - blend text, visualization and code
  - generate high quality reports that can be shared with the audience
- It supports a variety of output formats like PDF, HTML, among others
- Fully reproducible reports – each time you knit the analysis is run from the beginning
- Simple Markdown syntax for text

**Note:** Think of it as your "virtual notebook" that helps you code and document, all in one place!

# What does it entail?

- An R Markdown file is a simple text file with a `.rmd` extension
- Notice in the figure beside that it has 3 types of content,
  1. An *optional* YAML header enclosed within `---`
  2.  code chunks within `````
  3. Text with simple text formatting



```

1---  

2 title: "Viridis Demo"  

3 output: html_document  

4---  

5  

6```{r include = FALSE}  

7 library(viridis)  

8```  

9  

10 The code below demonstrates two color palettes in the  

11 [viridis](https://github.com/sjmgarnier/viridis) package. Each  

12 plot displays a contour map of the Maunga Whau volcano in  

13 Auckland, New Zealand.  

14  

15 ## Viridis colors  

16  

17```{r}  

18 image(volcano, col = viridis(200))  

19```  

20  

21## Magma colors  

22  

23```{r}  

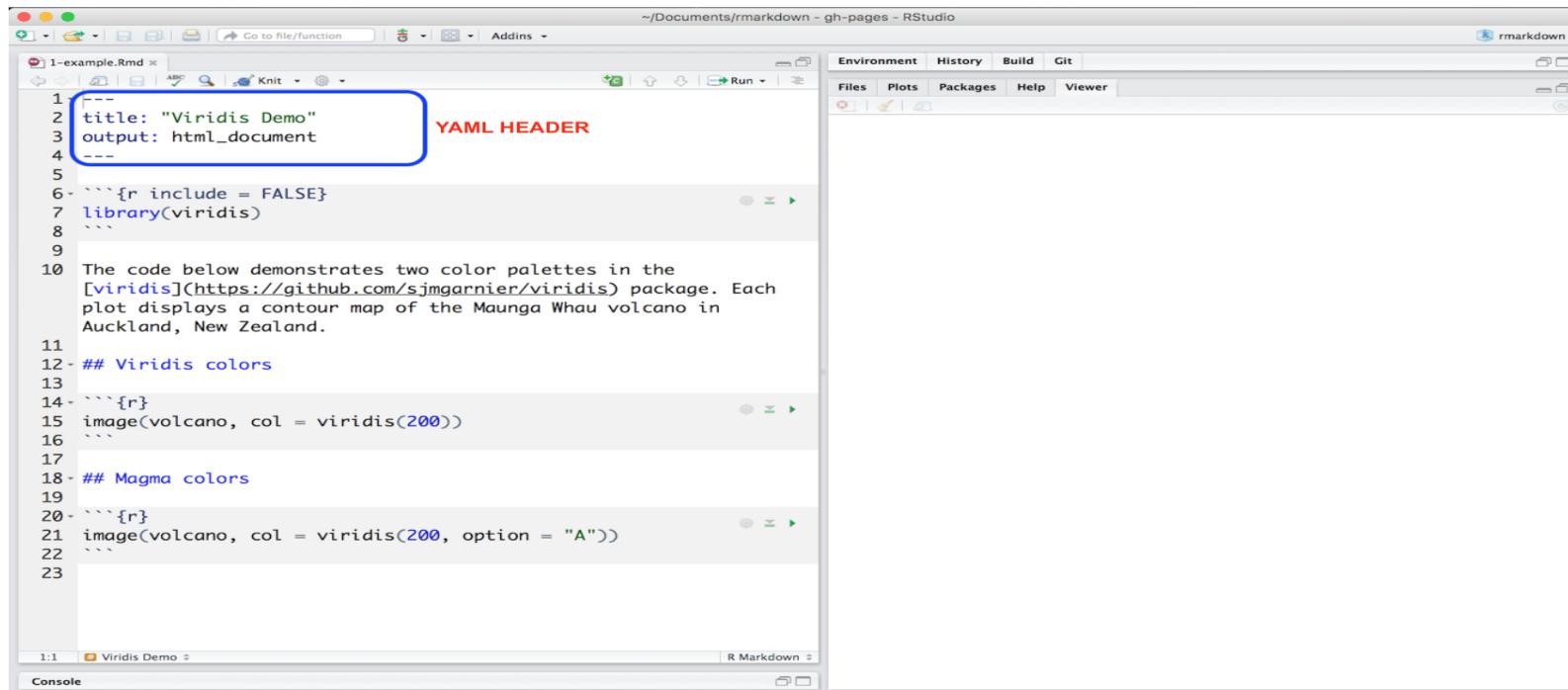
24 image(volcano, col = viridis(200, option = "A"))
25```

```

Figure: Typical R Markdown file

# What does it entail?

An *optional* YAML header enclosed within `---`



The screenshot shows the RStudio interface with an R Markdown file named "1-example.Rmd" open. The YAML header is highlighted with a red box and labeled "YAML HEADER". The code in the file is as follows:

```
1 ---  
2 title: "Viridis Demo"  
3 output: html_document  
4 ---  
5  
6 ````{r include = FALSE}  
7 library(viridis)  
8 ````  
9  
10 The code below demonstrates two color palettes in the  
11 [viridis](https://github.com/sjmgarnier/viridis) package. Each  
12 plot displays a contour map of the Maunga Whau volcano in  
13 Auckland, New Zealand.  
14 ## Viridis colors  
15 ````{r}  
16 image(volcano, col = viridis(200))  
17 ````  
18 ## Magma colors  
19 ````{r}  
20 image(volcano, col = viridis(200, option = "A"))  
21 ````  
22  
23
```

Figure: YAML header in a R Markdown file

# What does it entail?

## R code chunks within ...

The screenshot shows the RStudio interface with an R Markdown file open. The code editor on the left contains the following R code:

```
1 ---  
2 title: "Viridis Demo"  
3 output: html_document  
4 ---  
5  
6 ```{r include = FALSE}  
7 library(viridis)  
8 ```  
9  
10 The code below demonstrates two color palettes in the  
11 [viridis](https://github.com/sjmgarnier/viridis) package. Each  
12 plot displays a contour map of the Maunga Whau volcano in  
13 Auckland, New Zealand.  
14 ## Viridis colors  
15 ```{r}  
16 image(volcano, col = viridis(200))  
17 ```  
18 ## Magma colors  
19  
20 ```{r}  
21 image(volcano, col = viridis(200, option = "A"))  
22 ```  
23
```

Three specific code blocks are highlighted with blue rounded rectangles and labeled in red text:

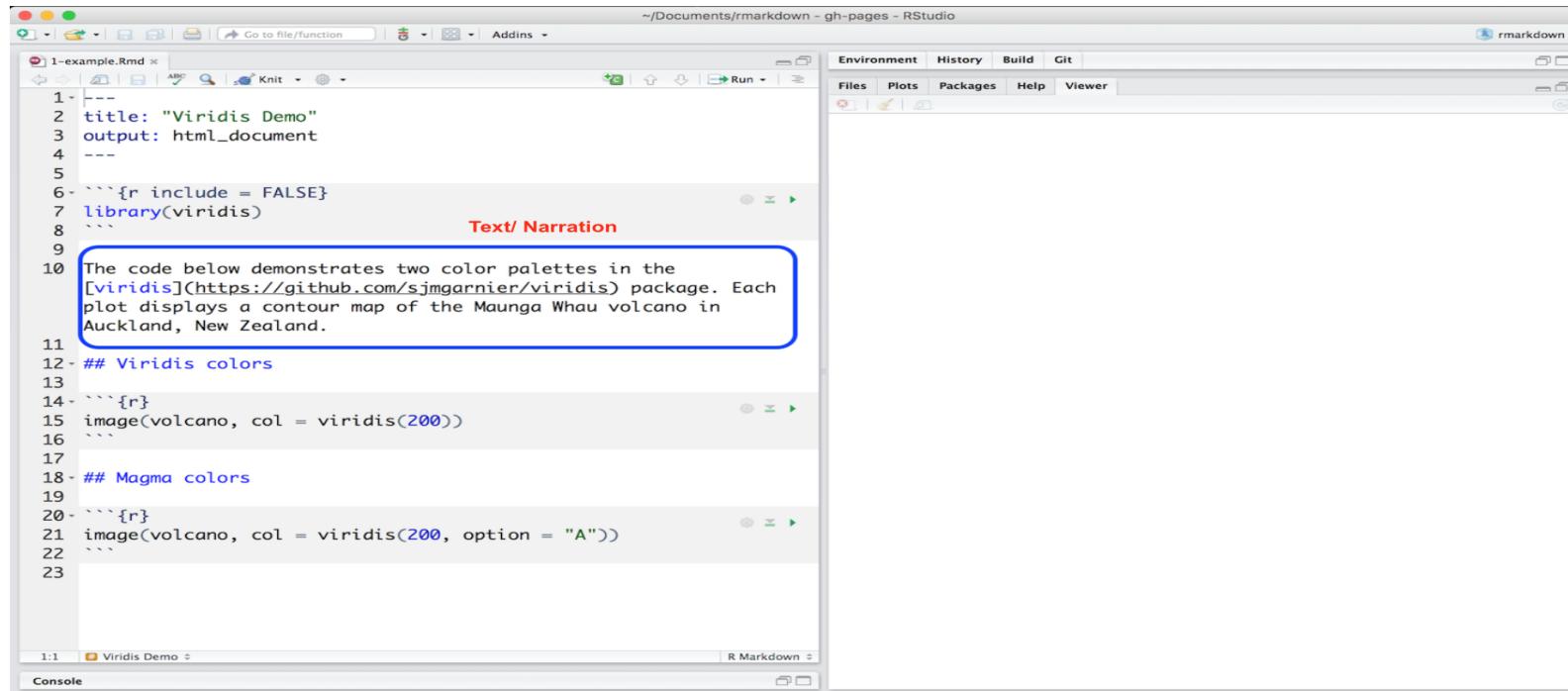
- Line 6: **Code Chunk 1**
- Line 15: **Code Chunk 2**
- Line 21: **Code Chunk 3**

The RStudio interface includes a top bar with tabs for Environment, History, Build, and Git, and a bottom navigation bar with tabs for Files, Plots, Packages, Help, and Viewer.

Figure: Code chunks in a R Markdown file

# What does it entail?

Text with simple text formatting



The screenshot shows the RStudio interface with an R Markdown file open. The code editor pane contains the following R Markdown code:

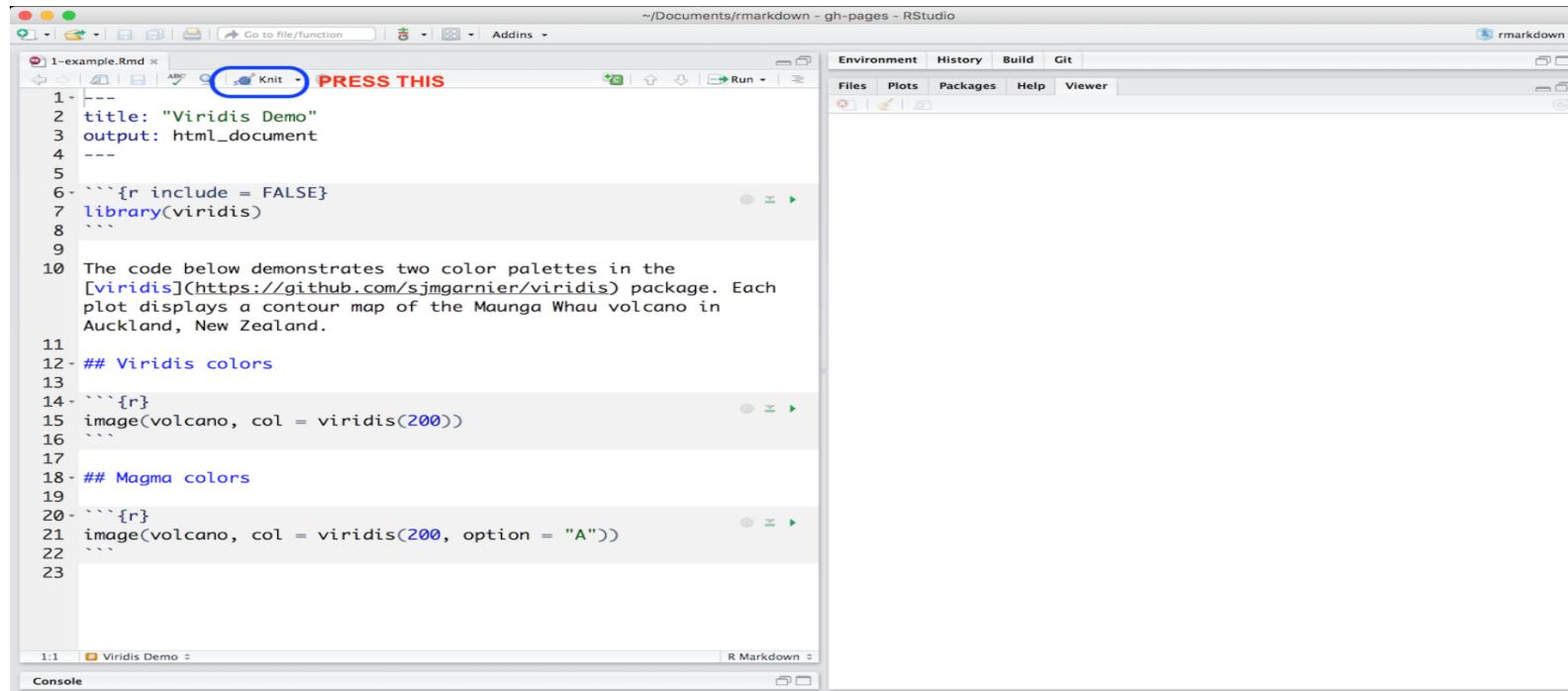
```
1---  
2title: "Viridis Demo"  
3output: html_document  
4---  
5  
6```{r include = FALSE}  
7library(viridis)  
8```  
9  
10The code below demonstrates two color palettes in the  
[viridis](https://github.com/sjmgarnier/viridis) package. Each  
plot displays a contour map of the Maunga Whau volcano in  
Auckland, New Zealand.  
11  
12## Viridis colors  
13  
14```{r}  
15image(volcano, col = viridis(200))  
16```  
17  
18## Magma colors  
19  
20```{r}  
21image(volcano, col = viridis(200, option = "A"))  
22```  
23
```

A red box highlights the narrative text from line 10, which is preceded by the text "Text/ Narration". The RStudio interface includes tabs for Environment, History, Build, Git, Files, Plots, Packages, Help, and Viewer.

Figure: Text in a R Markdown file

# Generating the report

To generate the output in the desired format, use the **Knit** button in posit Cloud IDE



```
1-example.Rmd
1 ---  
2 title: "Viridis Demo"  
3 output: html_document  
4 ---  
5  
6 ```{r include = FALSE}  
7 library(viridis)  
8 ```  
9  
10 The code below demonstrates two color palettes in the  
11 [viridis](https://github.com/sjmgarnier/viridis) package. Each  
12 plot displays a contour map of the Maunga Whau volcano in  
13 Auckland, New Zealand.  
14 ## Viridis colors  
15 image(volcano, col = viridis(200))  
16 ```  
17  
18 ## Magma colors  
19  
20 image(volcano, col = viridis(200, option = "A"))  
21 ```  
22  
23
```

Figure: Text in a R Markdown file

# How will we be using R Markdown?

- All the presentations in this course were prepared using R Markdown!!
- You will have to turn in all your submissions compiled using R Markdown 😊
- Worry not, you will be provided with a R Markdown document template to work with 😊
- The amount of scaffolding in the template will decrease over the course of the semester

# Thanks!

Slides created via the  packages:

xaringan  
gadenbuie/xaringanthemer.



Faculty of Arts  
& Social Sciences