Trivia: Know your knots!

A 

B 

C 

D 

E 

F 

1. Sheepshank
2. Cat’s Paw
3. Overhand
4. Half-hitch
5. Slipknot
6. Fisherman’s Bend

We will start at 9:00 Pacific / 10:00 Mountain / 11:00 Central / 12:00 Eastern / 13:00 Atlantic / 13:30 NFLD

Please take a moment to ensure that you have downloaded course materials for today, refresh your beverage, and / or network with us.

TDU

Introduction to R

For Public Health Investigations

Public Health Agency of Canada
Agence de la santé publique du Canada

Canada

The map

Day 1:

- Core-function coastline
- Syntax sound
- Clean data caves

Day 2:

- Markdown beach
- Automation bay
- Troubleshooting trail

What we heard

Designed by Wayhomestudio - Freepik.com

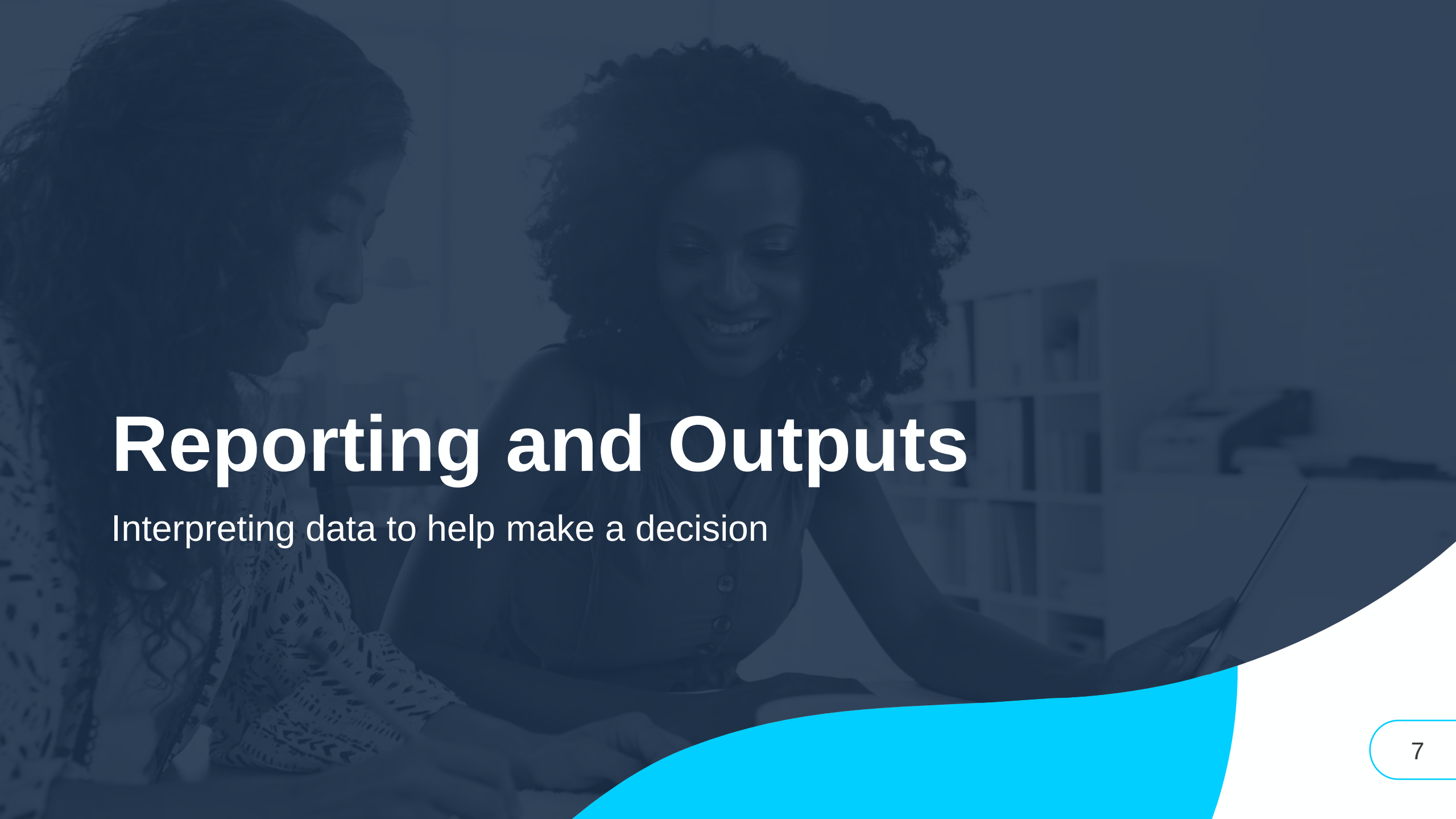
Exercise Debrief

Exercise 1:

COVID-19 line list

- Workspace setup
- Load data
- Clean the data including dates
- Create new variables
- Visualize and summarise the data
- Automate the results by writing a script

1. What was your favorite function or component of this activity?
2. Did anything surprise you? Did you do anything differently?
3. Do you foresee using any parts of this activity in your workplace? How?



Reporting and Outputs

Interpreting data to help make a decision

Reporting in a ‘perfect-world’

- Use the annotation tool to put a stamp on which of the following features of a reporting tool are important to you!

Automated

Attractive

Easy to update

Interactivity

Integrates data analysis

Multiple formats available

Tools for building reports in R

RMarkdown documents

- Several years of development, many extensions developed
- Primarily for R code
- Produces HTML, PDF, Word outputs natively
- Designed for use with RStudio
- Works best for single-document workflows

Quarto documents

- Successor to RMarkdown
- Built-in support for multiple coding languages (e.g., R, Python, Julia)
- More unified support
- Designed for use with RStudio as well as other IDEs (JupyterLab, VS Code, others)

R markdown – what is it?

- A tidyverse package: Rmarkdown

- The R markdown file format (.Rmd) is used to create documents in R (static or dynamic)

- Markdown is a formatting language that allows for documents to be written in plain text with formatting cues
  - R code can be written directly in the markdown document, or existing R scripts can be leveraged using the **source()** function and specifying the file pathway
  - Narrative text can be added and formatted nicely
  - Used to create reports and other data products in various formats

R markdown

- Three main components of an R markdown file (.Rmd):
  1. Metadata
     - YAML (‘YAML ain’t markup language’) header
     - Written between “---”
  2. Text
     - Markdown syntax
  3. Code
     - R code chunks

```
---
title: "Example"
author: "B. Hetman"
date: "2025-01-20"
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:

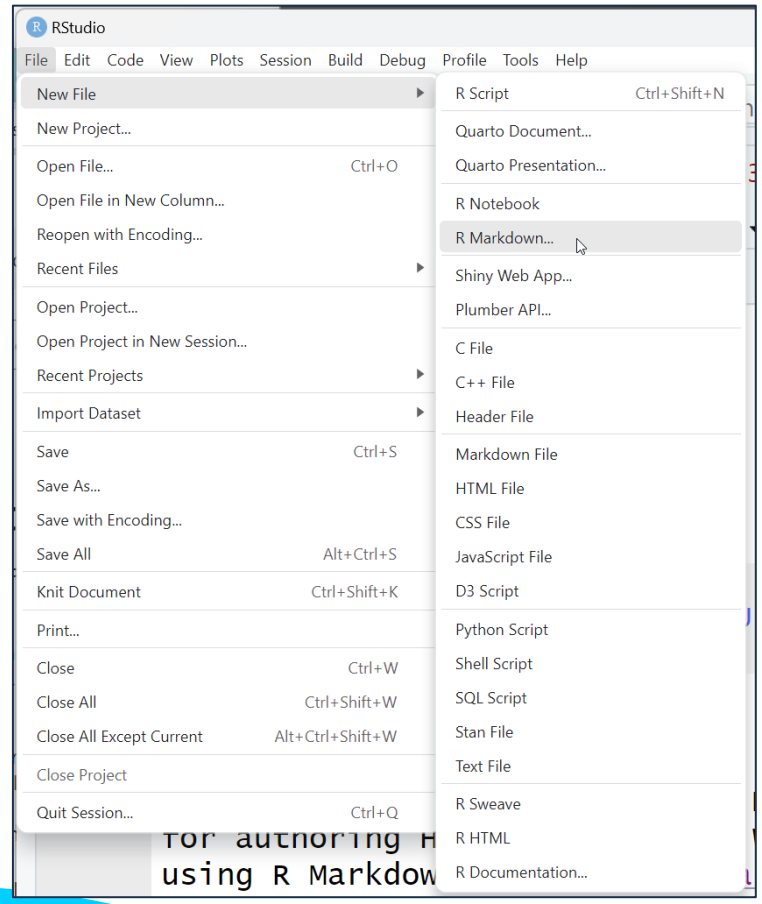
```{r pressure, echo=FALSE}
plot(pressure)
```
```

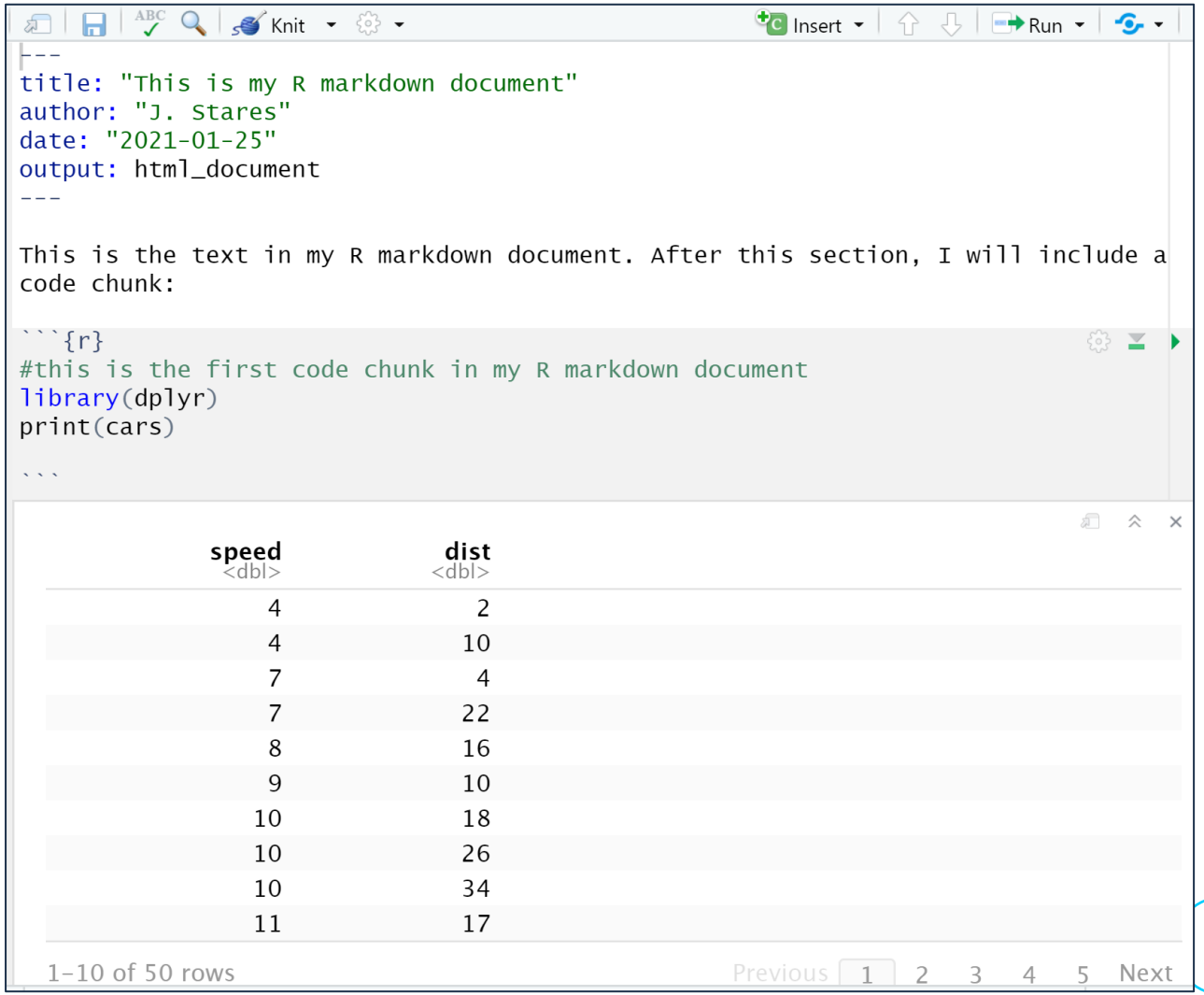
1

2

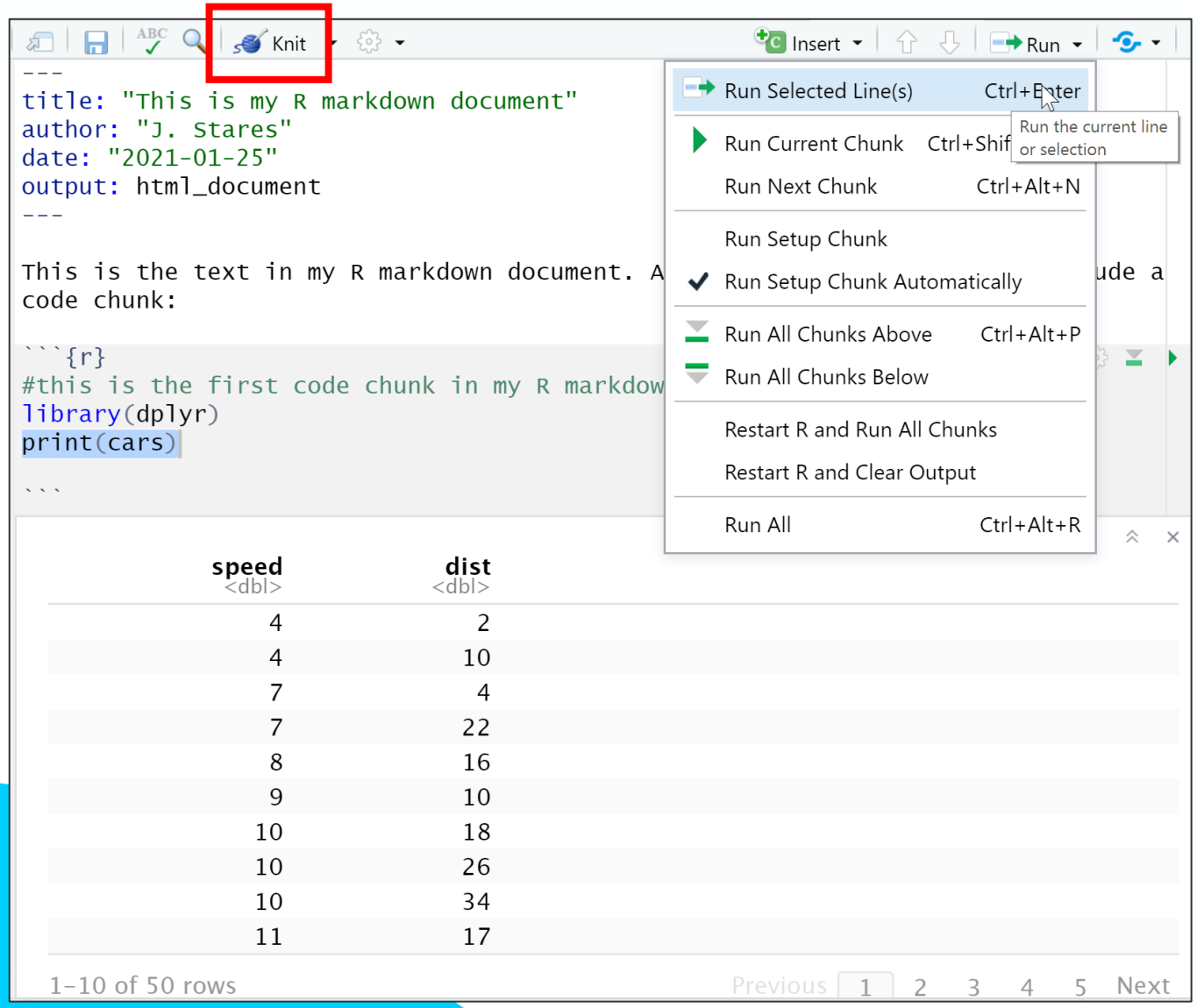
3

Creating an RMarkdown document





Testing and running your R markdown document



R markdown text format options

- Markdown allows for many text formatting options
- For example, specifying headers is done using the pound symbol (#)

R Markdown Reference Guide

Learn more about R Markdown at rmarkdown.rstudio.com
Learn more about Interactive Docs at shiny.rstudio.com/articles

Contents:

Syntax	Becomes
`Plain text`	Plain text
`End a line with two spaces to start a new paragraph.`	End a line with two spaces to start a new paragraph.
`*italics* and _italics_`	*italics* and *italics*
`**bold** and __bold__`	**bold** and **bold**
`superscript^2^`	superscript^2
`~~strikethrough~~`	~~strikethrough~~
`[link](www.rstudio.com)`	link
`# Header 1`	# Header 1
`## Header 2`	## Header 2
`### Header 3`	### Header 3
`#### Header 4`	#### Header 4
`##### Header 5`	##### Header 5
`###### Header 6`	###### Header 6
`endash: --`	endash: –
`emdash: ---`	emdash: —
`ellipsis: ...`	ellipsis: …
`inline equation: $A = \pi*r^{2}$`	inline equation: $A = \pi * r^2$
`image: ![](path/to/smallorb.png)`	image:

R markdown paragraph format

- Markdown allows several options to create formatted quotes, lists, tables (and more)

R Markdown Reference Guide

Learn more about R Markdown at rmarkdown.rstudio.com
Learn more about Interactive Docs at shiny.rstudio.com/articles

Contents:
1. **Markdown Syntax**
2. Knitr chunk options
3. Pandoc options

Syntax

```
***

> block quote

* unordered list
* item 2
    + sub-item 1
    + sub-item 2

1. ordered list
2. item 2
    + sub-item 1
    + sub-item 2

Table Header   | Second Header
-------------- | --------------
Table Cell     | Cell 2
Cell 3         | Cell 4
```

Becomes

block quote

- unordered list
- item 2
  - sub-item 1
  - sub-item 2

1. ordered list
2. item 2
   - sub-item 1
   - sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

R markdown outputs

- Depending on the options in your YAML header and in the setup of your code chunks, different outputs will be displayed in the final product.

This is my R markdown document

J. Stares

2021-01-25

This is the text in my R markdown document. After this section, I will include a code chunk:

```
#this is the first code chunk in my R markdown document
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
print(cars)
```

R markdown code chunk options (1)

- Entering different options into the curly braces following the three backticks will determine the treatment of the code chunk
- E.g.,
  - ```{r} is the default
  - ```{r eval=TRUE, echo=FALSE) will evaluate the code, but suppresses printing it to the final output
  - ```{r warning=FALSE} will suppress the printing of warnings in the output

R Markdown Reference Guide

Learn more about R Markdown at rmarkdown.rstudio.com
Learn more about Interactive Docs at shiny.rstudio.com/articles

Contents:
1. Markdown Syntax
2. **Knitr chunk options**
3. Pandoc options

Syntax	Becomes

Make a code chunk with three back ticks followed by an r in braces. End the chunk with three back ticks:
``{r}
paste("Hello", "World!")
``
	Make a code chunk with three back ticks followed by an r in braces. End the chunk with three back ticks:
	`paste("Hello", "World!")`
	`## [1] "Hello World!"`

Place code inline with a single back ticks. The first back tick must be followed by an R, like this `r paste("Hello", "World!")`.	Place code inline with a single back ticks. The first back tick must be followed by an R, like this Hello World!.

Add chunk options within braces. For example, `echo=FALSE` will prevent source code from being displayed:
``{r eval=TRUE, echo=FALSE}
paste("Hello", "World!")
``
	Add chunk options within braces. For example, `echo=FALSE` will prevent source code from being displayed:
	`## [1] "Hello World!"`

Learn more about chunk options at http://yihui.name/knitr/options

R markdown code chunk options (2)

Many more options available for creating clean, readable R markdown reports

Chunk options		
option	**default value**	**description**
Code evaluation		
child	NULL	A character vector of filenames. Knitr will knit the files and place them into the main document.
code	NULL	Set to R code. Knitr will replace the code in the chunk with the code in the code option.
engine	'R'	Knitr will evaluate the chunk in the named language, e.g. **engine = 'python'**. Run **names(knitr::knit_engines$get())** to see supported languages.
eval	TRUE	If **FALSE**, knitr will not run the code in the code chunk.
include	TRUE	If **FALSE**, knitr will run the chunk but not include the chunk in the final document.
purl	TRUE	If **FALSE**, knitr will not include the chunk when running **purl()** to extract the source code.
Results		
collapse	FALSE	If **TRUE**, knitr will collapse all the source and output blocks created by the chunk into a single block.
echo	TRUE	If **FALSE**, knitr will not display the code in the code chunk above it's results in the final document.
results	'markup'	If **'hide'**, knitr will not display the code's results in the final document. If **'hold'**, knitr will delay displaying all output pieces until the end of the chunk. If **'asis'**, knitr will pass through results without reformatting them (useful if results return raw HTML, etc.)
error	TRUE	If **FALSE**, knitr will not display any error messages generated by the code.
message	TRUE	If **FALSE**, knitr will not display any messages generated by the code.
warning	TRUE	If **FALSE**, knitr will not display any warning messages generated by the code.
Code Decoration		
comment	'##'	A character string. Knitr will append the string to the start of each line of results in the final document.
highlight	TRUE	If **TRUE**, knitr will highlight the source code in the final output.
prompt	FALSE	If **TRUE**, knitr will add > to the start of each line of code displayed in the final document.
strip.white	TRUE	If **TRUE**, knitr will remove white spaces that appear at the beginning or end of a code chunk.
tidy	FALSE	If **TRUE**, knitr will tidy code chunks for display with the **tidy_source()** function in the **formatR** package.

R markdown: Let’s review (1)

- What sorts of documents can be created using R markdown?
  - Pdf, word, html, ppt, dashboard, webpages and more

- Is the # needed to add narrative text to your R markdown file?
  - No, the # isn’t needed to add narrative text – it is used for formatting
  - Note # still required to comment R code within code “chunks”

- Why would you use R markdown?
  - Reproduce your work at the click of a button, export results as a report or link to dashboard, create a transparent record of your work, etc.

R markdown: Let’s review (2)

- What is the header used to open a chunk of R code in a markdown file?
  - ```{r}

- How do you supress printing of R code in your rendered file?
  - In the code chunk header: echo=FALSE

```{r, echo=FALSE}
```

- How do you create an ordered list in a markdown document?
  - Just number the text:
    1. Item 1
    2. Item 2, etc.

- How do you render and export your document?
  - Knit the document: 

Considerations for Programming

Save yourself from yourself!

Programming tips

Do's	Do Not's
Create a header that includes: • Date the code was written and date it was last modified • The author and person who last modified the code • Purpose of the code • Associated datasets and location • Pertinent notes (e.g., decisions)	Leave code and syntax documents in such a way that it is unclear what they are for, when they were created, or who created them (this is especially important for tasks which are run regularly but not frequently)
If using one script document, chunk the code by various stages of the project (e.g., loading data, processing data, analysing data, summarising data, visualising data, reporting and data export)	Leave code in such a state that it is unclear when to run which code block
Write brief comments throughout explaining what code blocks or complicated code is doing	Leave code in such a state that is unclear what they are meant to do
Revise code such that it can be run all at once, especially if automating tasks	Leave code in your script that does not work or that is not needed

Automation: When is it appropriate?

Appropriate:

- Conducting an analysis that will be repeated regularly
- Conducting an analysis that will be revisited at a later date
- Other?

Not appropriate

- Conducting an analysis which will be run once only
- Handing over a “one-off” data analysis to another individual
- Other?

Ten considerations for data analysis reference

- Use this handy list to help you plan your project at the outset
- Checklists aren't a replacement for critical thinking skills
- You may need to add other considerations (e.g., data privacy)
- Or you may need to skip considerations depending on your objective

Read Read data into R	Consider (1) the type of data, (2) how they are stored, and (3) the size of the dataset.
Format Format the data	Long or wide format? Plan the analysis keeping in mind what your needs are. Long: summarising, visualising. Wide: human-readable, tables.
Select Select portions of the dataset	Decide what data you'll need in advance from the dataset: subsets vs. filtered dataset. Consider the pros and cons of the approach.
Modify Create and modify variables	Consider: (1) what variables you need, (2) calculations you will perform, and (3) algorithms needed to create the variables.
Link Conduct data linkage(s)	Assign a key variable(s) that you will link datasets on. Consider the join type you will need: inner, left, right, full, semi, or anti join.
Analyse Analyse the data	Consider grouping data meaningfully in performing calculations and obtaining descriptive statistics. Evaluate outputs and interpret summaries.
Visualise Visualise the data	Use tables or charts, and consider the best way to display information. Often, the simpler the better. What are the data telling you?
Report Report the findings	Communicate the findings to those who need the information created from your work. Consider the audience you are communicating with.
Celebrate	Preferably with cake!
Review Review, revise, and re-do as needed	Often, we can go back and simplify/optimise. This is important especially if the code will be used again.

Getting the most from Google

- It is an extraordinary human that can sit and write R code like they are writing a letter

- Google is your friend!
  - As are stackoverflow.com, tidyverse.org, rdocumentation.org, cookbook-r.com, etc.

- Tip in framing web search: software + version + key terms

If you can’t tie knots…

Photo by Miguel A. Amutio on Unsplash

Photo by Pascal van de Vendel on Unsplash

…tie lots!

Break

- Take a few minutes for yourself
  - Too many scraps of paper on your desk?
  - Need a mindfulness activity suggestion?
  - Build yourself a flotilla of paper ships: https://www.wikihow.com/Make-a-Paper-Ship

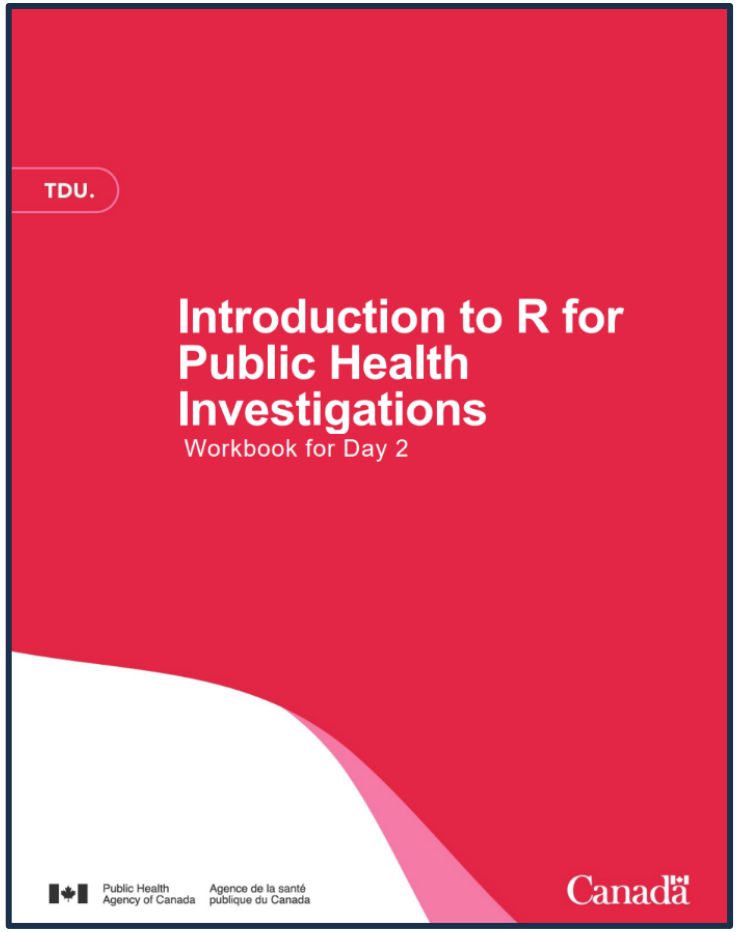
Please return by:
[TIME] Pacific / [TIME] Mountain / [TIME] Central / [TIME] Eastern / [TIME] Atlantic / [TIME] NFLD

Independent Study and Drop-In Office Hours

Practice makes perfect

Exercise 2

- Scenario: You've been mobilized to support an investigation of a tuberculosis outbreak in a remote, northern area.
- For this request you will:
- Setup your workspace
- Clean and process data
- Create descriptive epi figures and summary tables relevant to the outbreak investigation
- Create a social network diagram
- Build an automated report using R markdown



Approach

- We recommend:
  - **Novice users:** Use the workbook and R script(s) provided on GitHub as a guide. Run the available scripts and prioritize your understanding what each chunk of code and functions used are doing. Do not worry about being able to write or debug code.
  - **Beginner/Intermediate users:** R code is provided as a screen capture image in the workbook. You should have sufficient understanding of coding to get a general sense of what the code is doing by reading it or doing a little research. We would like you write the code out from the guide as you progress through the scenario. Cross reference to the R script(s) provided on GitHub if you encounter any tangly problems.
  - **Advanced users:** We encourage you to try writing your own code where you like and contrast it with the code used for the exercise, and to help your peers as questions arise. Cross reference to the R script(s) provided on GitHub if you encounter any tangly problems.

Independent Study and Drop-In Office Hours

- We will walk through the first steps of the exercise to ensure that everyone is able to get started (optional attendance)
- You are free to stay in the virtual classroom or leave while you work through the exercise
- We'll be here in the virtual classroom to answer your questions as they arise
- Please return by 12:15 Pacific / 13:15 Mountain / 14:15 Central / 15:15 Eastern / 16:15 Atlantic / 16:45 NFLD

A gentle reminder to take a break and stretch

Please return by:

- **12:15 Pacific**
- **13:15 Mountain**
- **14:15 Central**
- **15:15 Eastern**
- **16:15 Atlantic**
- **16:45 NFLD**

Questions?

Wrap-up

Reminder

- Complete exercise 2 as we will debrief when we meet tomorrow
- Please reach out to your course facilitators if you have questions

Feedback for day 2

- Please take a moment to answer a few questions
- https://www.slido.com/
- #IntroR2025