Module 3: R

Professional Skills

Amy Farrow

Data Sciences Institute, University of Toronto

2022-10-17

Overview

- Presenting Data Science
- Presenting Your Work in R
- Managing Data Science Projects

Presenting Data Science

Challenges

- A target audience that will likely not have an equivalent technical understanding to your own
- Communicating the limitations while promoting your work
- Keeping it interesting
- Including the appropriate level of persuasion

[redacted]

Strategies

- Don't just think in terms of performance and technical ability of your analysis and models. Consider: -- Are your models believable? -- What evidence have you offered? How have you built trust?
- Don't overcomplicate it -- What details are necessary to understanding the core goals, abilities, and limitations of your project? Which are not?
- Tailor your message -- Who are you presenting to? What level of knowledge can you assume?
- Always have takeaway messages -- In a presentation where some viewers might get confused along the way, a strong structure and final message can keep everyone on track

Practical Considerations: Presenting Your Work in R

Presenting Data Tables

There are many libraries to format tables for readable and professional-looking outputs. Some only work for certain kinds of outputs (html, pdf, or docx), so you may have to pick a library that is the best fit for you.

`kableExtra` is a library for formatting table output in html and pdf.

A table without formatting looks like this:

```r
my_table <- tibble(categoryA = c(1,2,3, 4),
       categoryB = c("one", "two", "three", "four"),
       other = c(14.3, 182.5, 54.0, 33.1))
my_table
```

```
## # A tibble: 4 × 3
##   categoryA categoryB other
##       <dbl> <chr>     <dbl>
## 1         1 one        14.3
## 2         2 two       182.
## 3         3 three      54
## 4         4 four       33.1
```

With formatting:

```
my_table %>%
  kable(caption = "My Table", booktabs = TRUE,
                col.names = c("A", "B", "Other")) %>%
  pack_rows("Run 1", 1, 2) %>%
  pack_rows("Run 2", 3, 4) %>%
  add_header_above(c("Categories" = 2, " " = 1))  %>%
  kable_styling()
```

My Table

Categories		
A	B	Other
Run 1		
1	one	14.3
2	two	182.5
Run 2		
3	three	54.0
4	four	33.1

Formatting Reports

The package `bookdown` can be used to cross-reference figures and tables, add citations, create a table of contents, and more.

For the output to work correctly, you want to have blank lines in your markdown in between parts: i.e. between a paragraph and the next paragraph, a paragraph and a code chunk, a code chunk and a paragraph, or a code chunk and the next code chunk.

A typical header for such a report would look like:

Breaking down the header

```
---
title: "TITLE"
subtitle: "SUBTITLE"
author: "YOUR NAME"
date: "`r format(Sys.time(), '%d %B %Y')`"
output:
  bookdown::pdf_document2:
    toc: yes
abstract: "ABSTRACT"
bibliography: references.bib
---
```

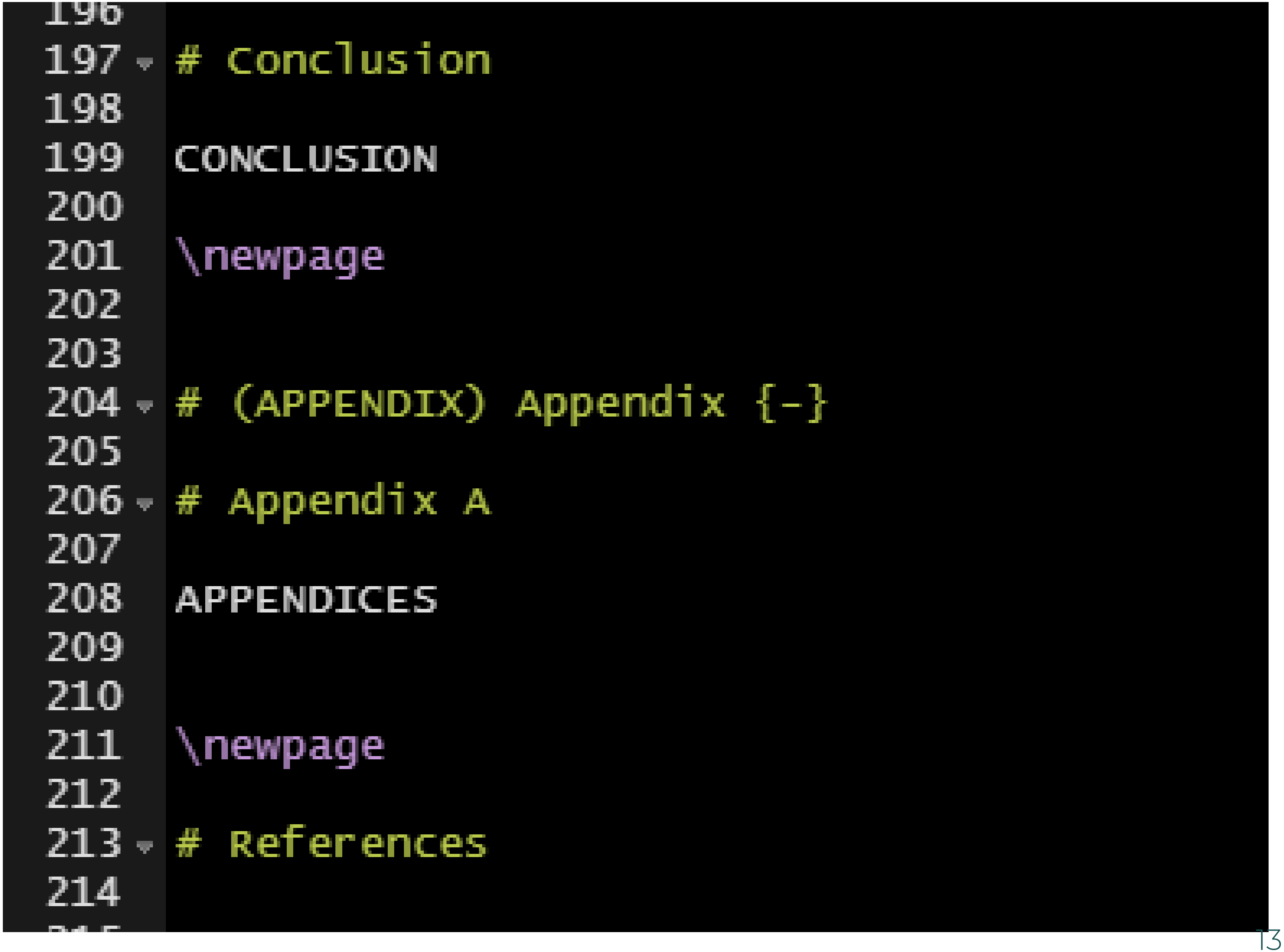
- The title, subtitle, author, and abstract can be inserted in the quotes.
- The date is automatically generated based on the system time and date.
- The output for the report is pdf_document2, a `bookdown` format.
- A table of contents will be included in the output.
- The bibliography will be created based on a bib file called references.

Breaking down the setup chunk

```
knitr::opts_chunk$set(
  echo = FALSE,     # hide source code in output
  message = FALSE, # hide messages from  code in output
  warning = FALSE  # hide warnings from code in output
  )

library() # load libraries here, including bookdown
```

- We set all the chuns to hide code, warnings, and messages by default, while still showing the code output.
- All necessary libraries are loaded in the chunk.

A typical end of a report would look like this:



```

# Conclusion

CONCLUSION

\newpage


# (APPENDIX) Appendix {-}

# Appendix A

APPENDICES


\newpage

# References

```

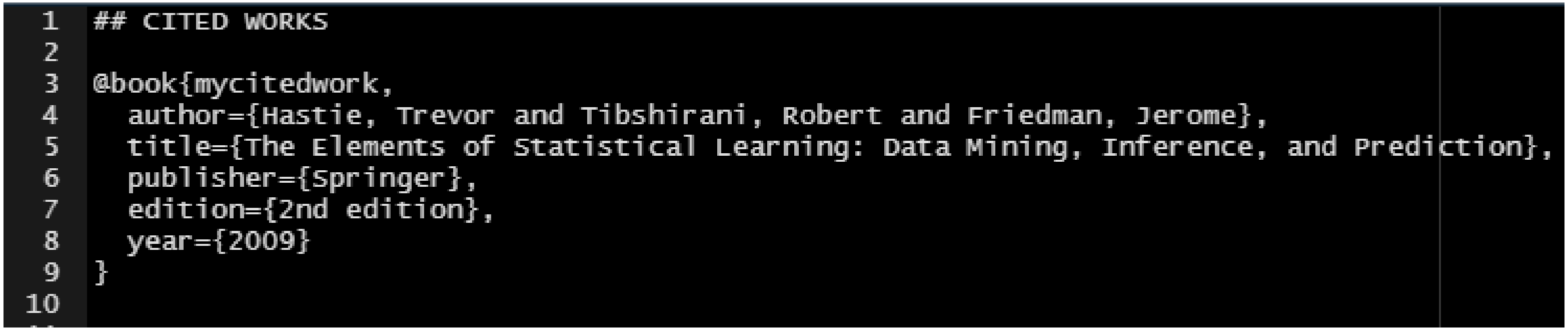
Breaking down the end of the report

```
# Conclusion

CONCLUSION

\newpage


# (APPENDIX) Appendix {-}

# Appendix A

APPENDICES


\newpage

# References
```

- Our last report section is the conclusion. We want a page break in between the conclusion and the appendices.
- We also want a page break before our references section. References will be automatically attached to the end of the report by `bookdown`.

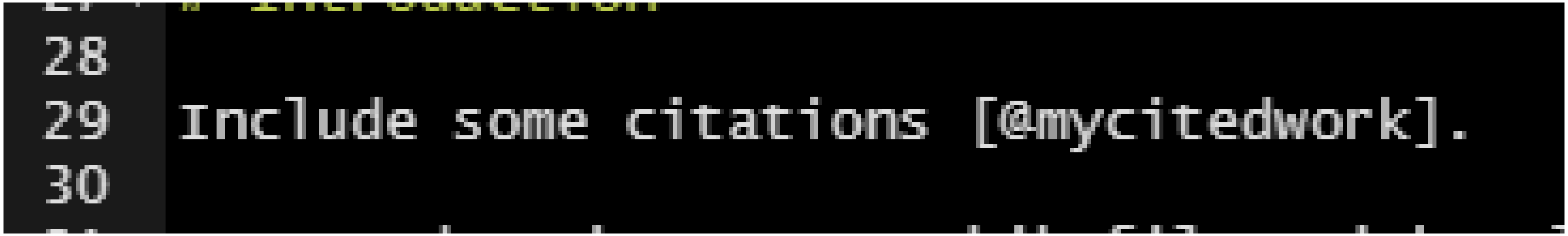
Referencing

Each work that you wish to cite will need to be present in the .bib file, where it will be given a unique nickname:



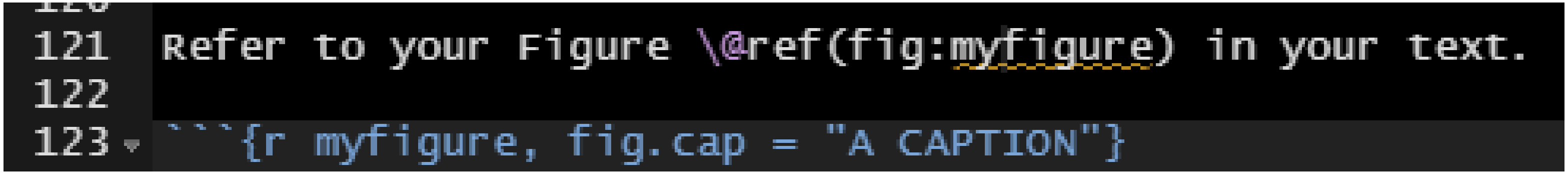
```
## CITED WORKS

@book{mycitedwork,
  author={Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome},
  title={The Elements of Statistical Learning: Data Mining, Inference, and Prediction},
  publisher={Springer},
  edition={2nd edition},
  year={2009}
}

```

To reference it in your text, you will refer to that nickname again, in the form `@mycitedwork` or `[@mycitedwork]`:



```

Include some citations [@mycitedwork].

```

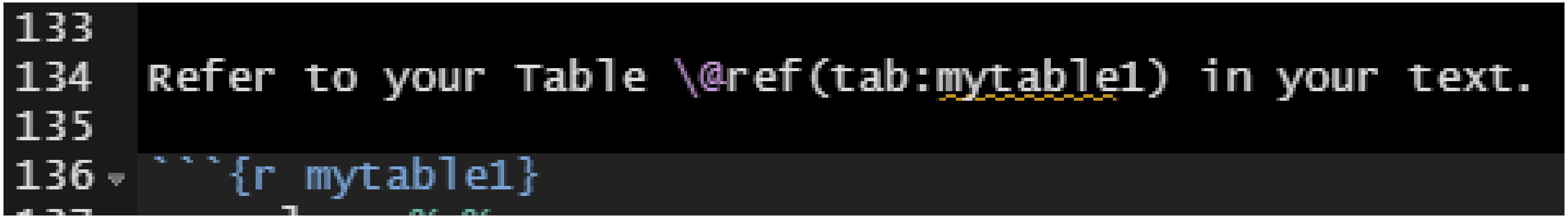
Cross-referencing figures



~~~
Refer to your Figure \@ref(fig:myfigure) in your text.

```{r myfigure, fig.cap = "A CAPTION"}
~~~

- Each chunk with a figure (ex. ggplot graph) will require a name (alphanumeric, excluding spaces and underscores) and a caption ("A CAPTION"). These must be unique to each figure.
- To reference the figure, you can use the code `\@ref(fig: )`, with the figure name included in the brackets.

Cross-referencing tables



~~~
Refer to your Table \@ref(tab:mytable1) in your text.

```{r mytable1}
~~~

- Each chunk with a table will also need a unique name.
- In the body of your code chunk, you also need to give the table a caption. Exactly how this is accomplished will depend on what table formatting package you are using.
- To reference the table, you can use the code `\@ref(tab: )`, with the table name included in the brackets.

Managing Data Science Projects

Objectives

Before embarking on a project, the team should be able to provide answers to each of the following:

1. Regulatory requirements
2. Frequency of model updating
3. Consequences of being wrong
4. Volume of data
5. Method for users to access results
6. Level of access/connection between the data science team and end users

Skills

1. Do the skills currently exist?
2. Are the people with the skills available?
3. What are the consequences of failing to complete?
4. How urgent is the project?
5. Would it be difficult to hire a temp?

Data

1. Has the team worked with this data before?
2. What is the data provenance?
3. Would you benefit from more data, and would acquiring it be feasible?
4. Do you have permissions required to use the data?
5. Will the data be refreshed frequently enough for your model?

Discussion questions

1. What do you think is the biggest challenge in communicating data science to non-data scientists?
2. How can you balance giving a thorough presentation and making the work accessible?
3. What questions would you add to the data science project checklist?