



# Intro to R Markdown

---

Biological statistics III

Elsa Fogelström, [elsa.fogelstrom@gmail.com](mailto:elsa.fogelstrom@gmail.com)

6 December 2021

# Today's session

Learn the basics of making a report using R Markdown

Produce neat, reproducible reports directly from your code

## Reproducibility & repeatability

- Code, notes and output in one document

More on repeatability and reproducibility [here](#) and [here](#)

# R Markdown



Save and execute R code

Add text with simple formatting to produce

- Documents (reports, manuscripts ...)
- Presentations
- Web pages
- Books

directly from your code and data



<https://rmarkdown.rstudio.com/>

# Prerequisites

An editor, e.g. [RStudio](#), to create and edit R Markdown document

The `rmarkdown` package

```
install.packages("rmarkdown",  
dependencies = TRUE)
```

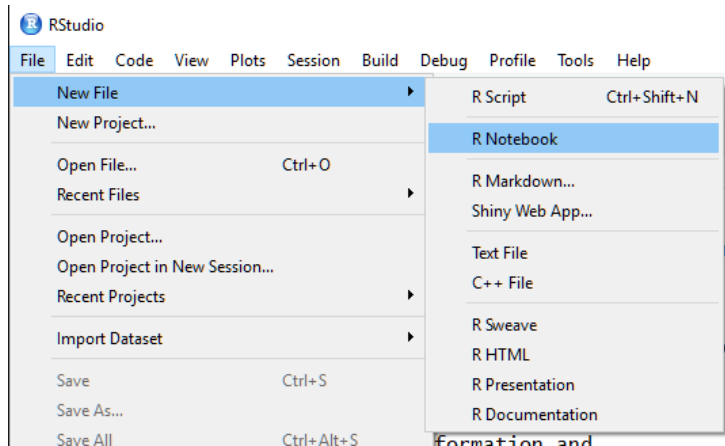
LaTeX, e.g. [TinyTeX](#), to produce PDF documents

```
install.packages("tinytex")  
tinytex::install_tinytex()
```

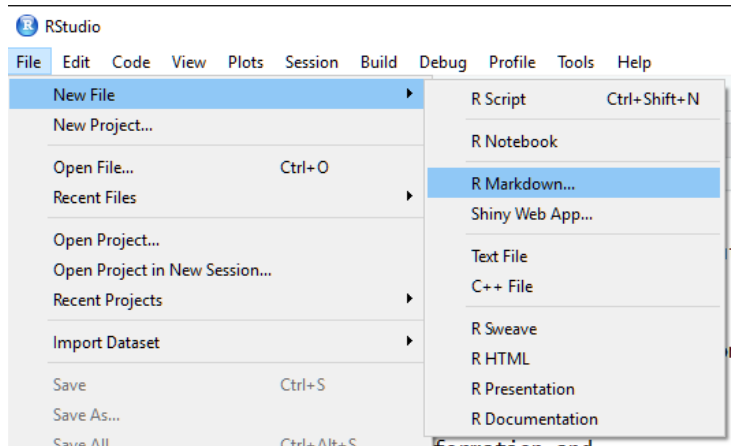
Installing `tidyverse` will install both RMarkdown and TinyTeX. Tidyverse contains many useful packages for data management and visualization.

```
install.packages("tidyverse",  
dependencies = TRUE)
```

# Creating an R Markdown document

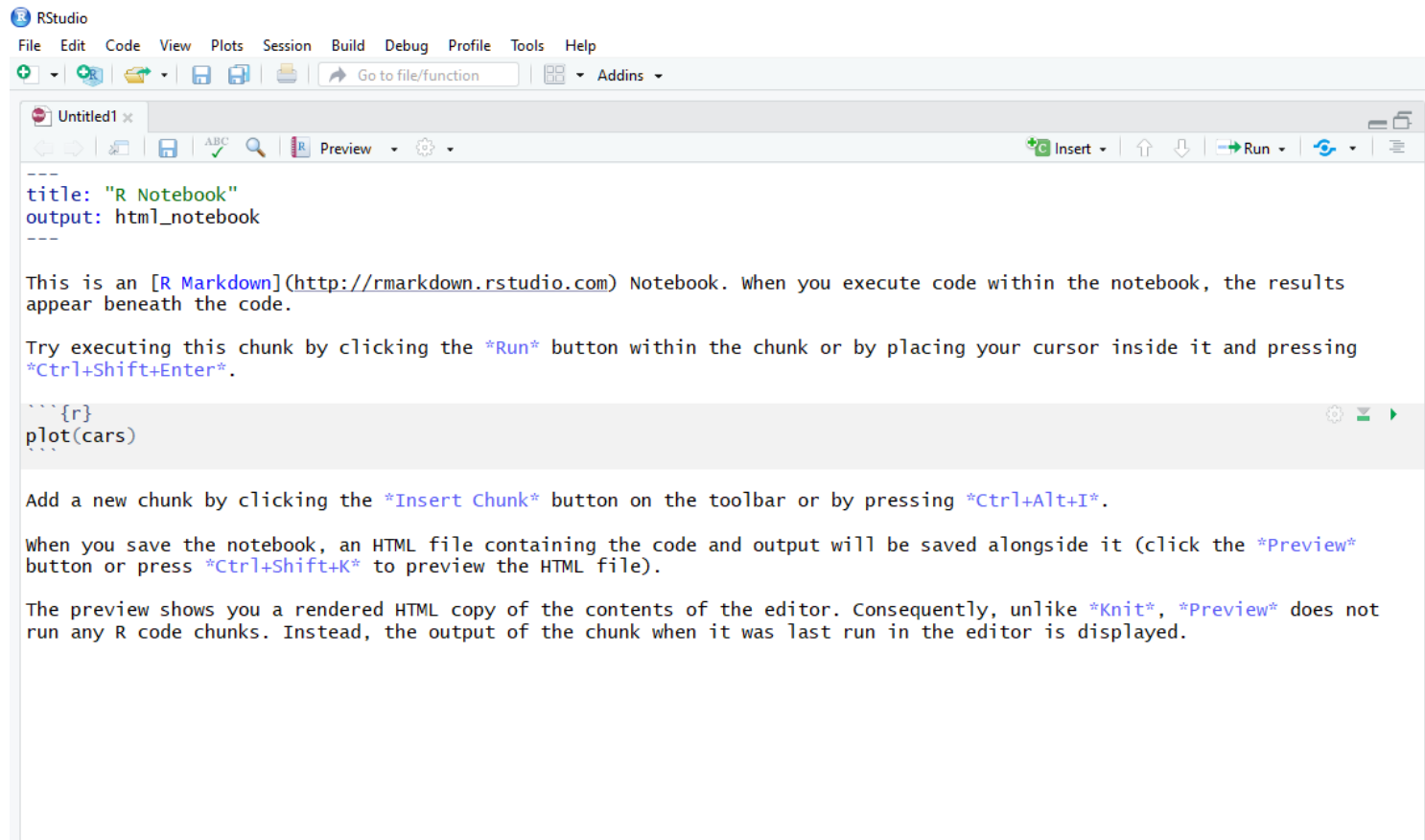


Create an basic .Rmd document  
.rmd documents are displayed in  
"notebook-mode" per default

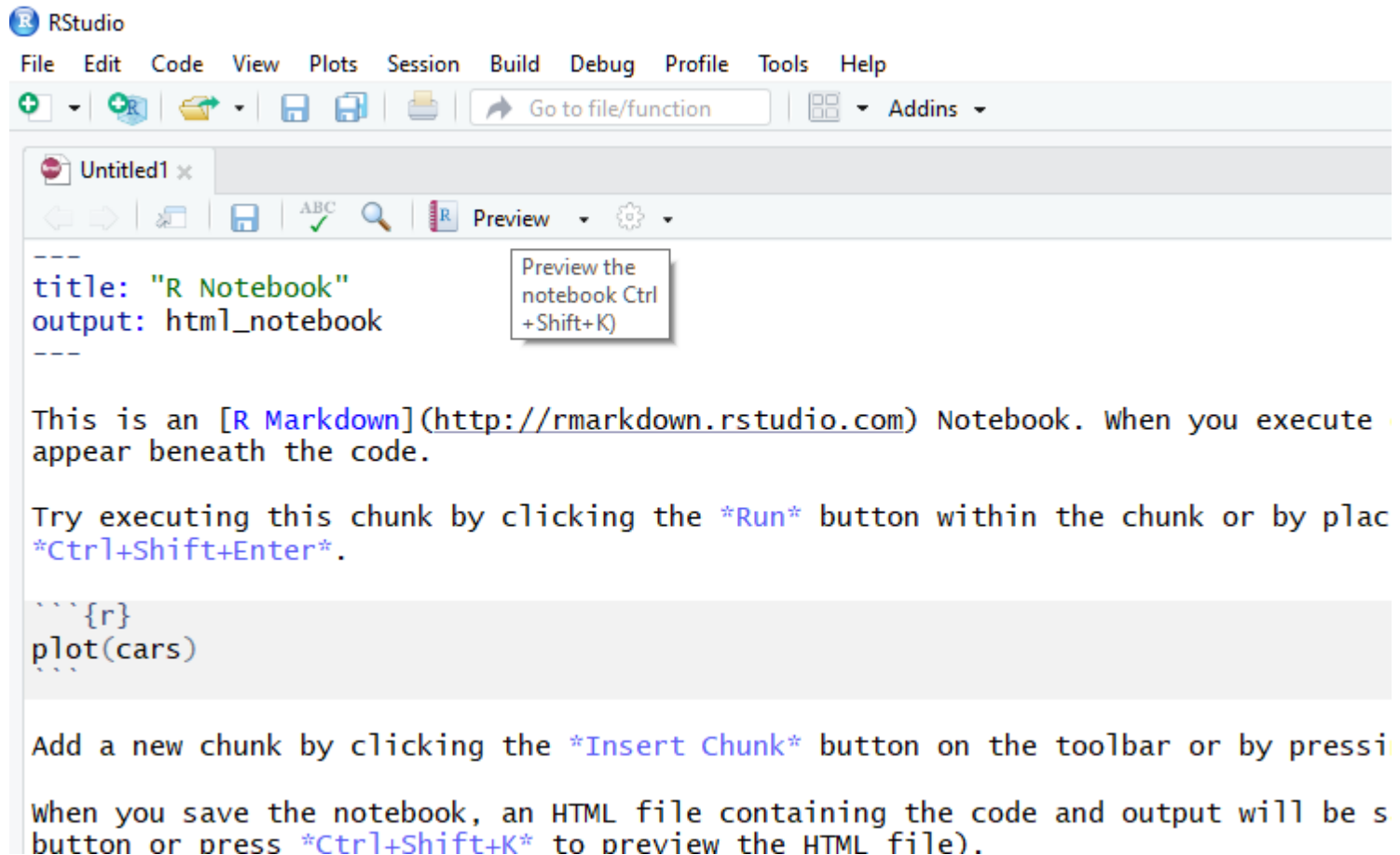


Create .Rmd documents, presentations  
and more

# Creating an R Markdown document



# Previewing an R Markdown document



The screenshot shows the RStudio application window. The title bar reads 'RStudio'. The menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains icons for creating a new file, opening a file, saving, and a search bar labeled 'Go to file/function'. Below the toolbar, the file name 'Untitled1' is shown. The editor area contains the following R Markdown code:

```
---  
title: "R Notebook"  
output: html_notebook  
---  
  
This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute  
appear beneath the code.  
  
Try executing this chunk by clicking the *Run* button within the chunk or by plac  
*Ctrl+Shift+Enter*.  
  
````{r}  
plot(cars)  
````  
  
Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressi  
When you save the notebook, an HTML file containing the code and output will be s  
button or press *Ctrl+Shift+K* to preview the HTML file).
```

A tooltip for the 'Preview' button (represented by an R logo) is displayed, showing the text: 'Preview the notebook Ctrl + Shift + K'.

# Previewing an R Markdown document

The screenshot displays the RStudio interface with a project named 'Notebook.Rmd'. The editor window on the left contains R Markdown code. The first chunk has a title 'R Notebook' and output format 'html\_notebook'. The second chunk contains a paragraph about R Markdown and a code chunk `plot(cars)`. The third chunk contains instructions on how to add a new chunk and preview the document. The right-hand pane shows the 'Environment' tab, which is empty, and the 'Viewer' tab, which displays a rendered HTML preview of the notebook. The preview includes the title 'R Notebook', the introductory text, and the `plot(cars)` code chunk. The 'Code' button in the top right of the preview is active.

**RStudio Interface:**

- Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for saving, running, and inserting code chunks.
- Environment Panel:** Shows 'Global Environment' with an empty table of variables.
- Viewer Panel:** Displays the rendered HTML output of the R Markdown document.

**R Markdown Document Content:**

```
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When
you execute code within the notebook, the results appear beneath the
code.

Try executing this chunk by clicking the *Run* button within the chunk
or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.



```
{r}
plot(cars)
```



Add a new chunk by clicking the *Insert Chunk* button on the toolbar
or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and
output will be saved alongside it (click the *Preview* button or press
*Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the
editor. Consequently, unlike *Knit*, *Preview* does not run any R code
chunks. Instead, the output of the chunk when it was last run in the
editor is displayed.
```

**Preview Content:**

## R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
plot(cars)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

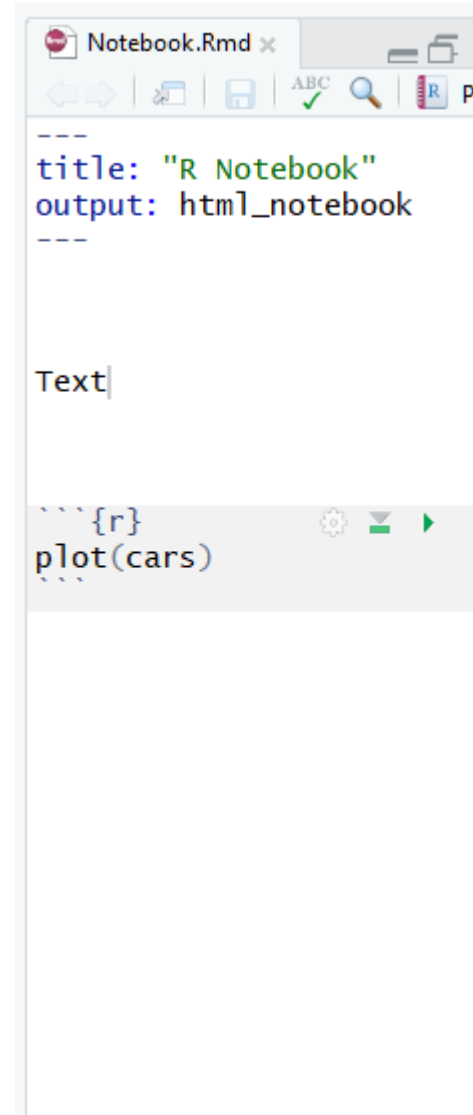


# R Markdown (.rmd) document components

**Metadata**

**Text**

**Code chunks**



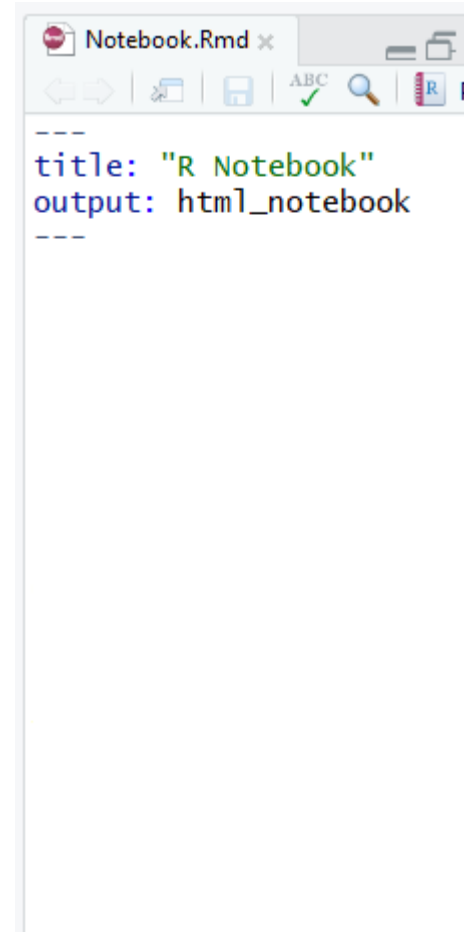
# Metadata

Document specific settings (author, date, output format ...)

Separated with 3 dashes ( --- )

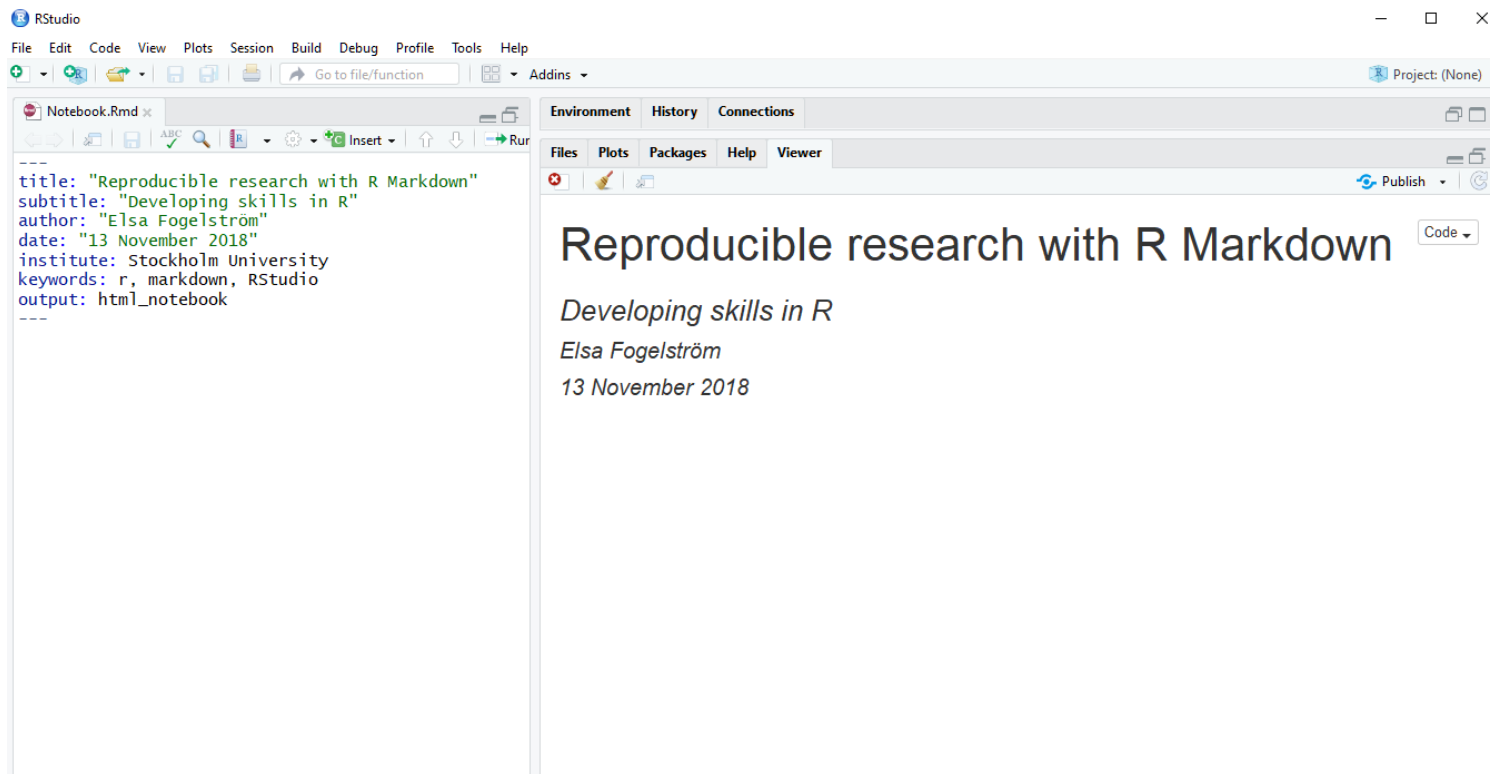
Syntax = YAML

Often referred to as "YAML metadata",  
"YAML header", "YAML frontmatter" ...



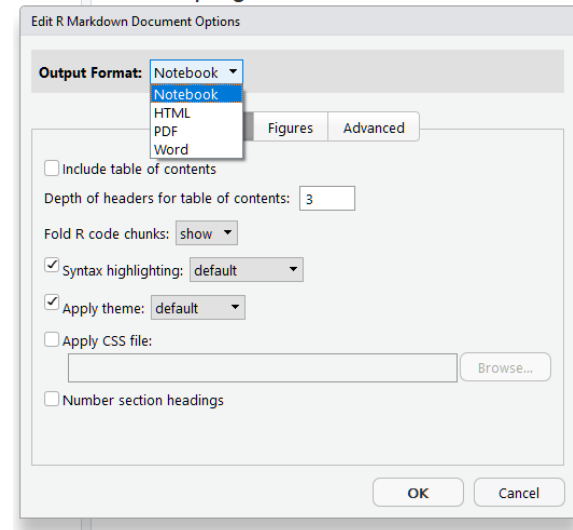
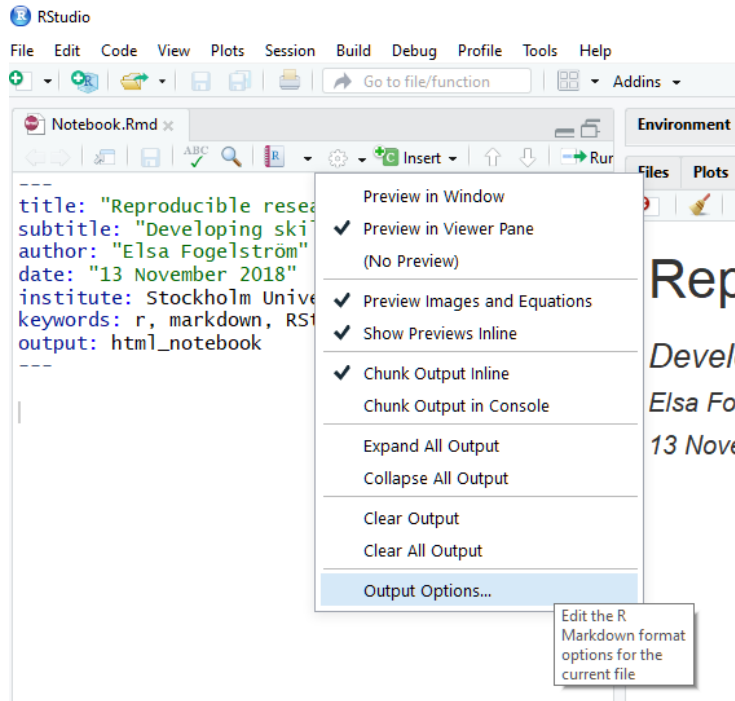
# Editing metadata

Title, author(s), date



# Editing metadata

Output options (doesn't cover all output formats):



# Editing metadata

## Output formats

github\_document

html\_document

latex\_document

md\_document

odt\_document

pdf\_document

word\_document

beamer\_presentation

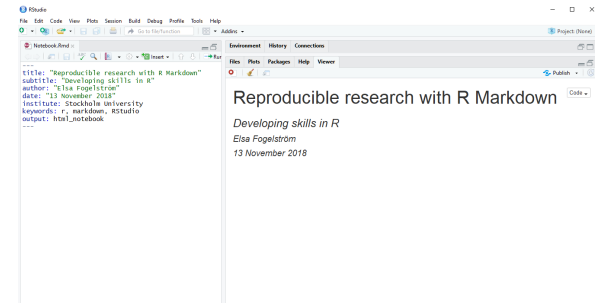
ioslides\_presentation

powerpoint\_presentation

rtf\_document

slidy\_presentation

and more from templates



# Text

**Markdown**, a simple formatting language ([official website](#))

Plain text with simple formatting

Other languages are also supported (e.g. HTML, LaTeX ...)

# Text formatting

To make sure that a line break is made, **use double space**

`*italics*`

*italics*

`**bold**`

**bold**

`# header1`

**header1**

`## header2`

**header2**

`### header3`

**header3**

# Text formatting: Lists

\* this is

\* a list

- this is

- another list

1. this is a

2. numbered list

- this is

- a list

- this is

- another list

1. this is a

2. numbered list

**remember double space**



# Text formatting: Mathematical annotation (LaTeX)

inline mathematical annotation:

```
Eqn. 1:  $\frac{x}{x^2}$ 
```

Eqn. 1:  $\frac{x}{x^2}$

centered mathematical annotation:

```

$$Y \sim \beta_0 + \beta_1 * x + \epsilon$$

```

$$Y \sim \beta_0 + \beta_1 * x + \epsilon$$



# Code

Code is included in *chunks* separated by three backticks

A chunk:

```
```{r}  
plot(iris$Petal.Length, iris$Petal.Length)  
```
```



# Code

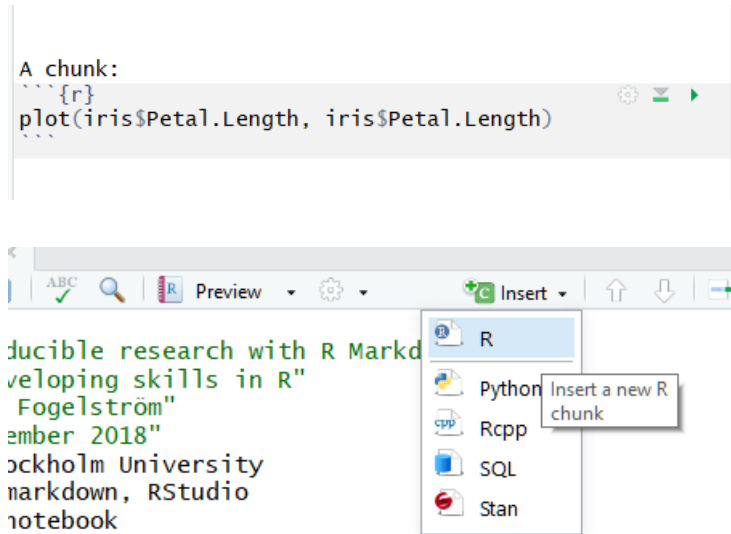
Code is included in *chunks*  
separated by three backticks  
( ````` )

"insert"/"R"

keyboard shortcuts:

*Ctrl+Alt+i*

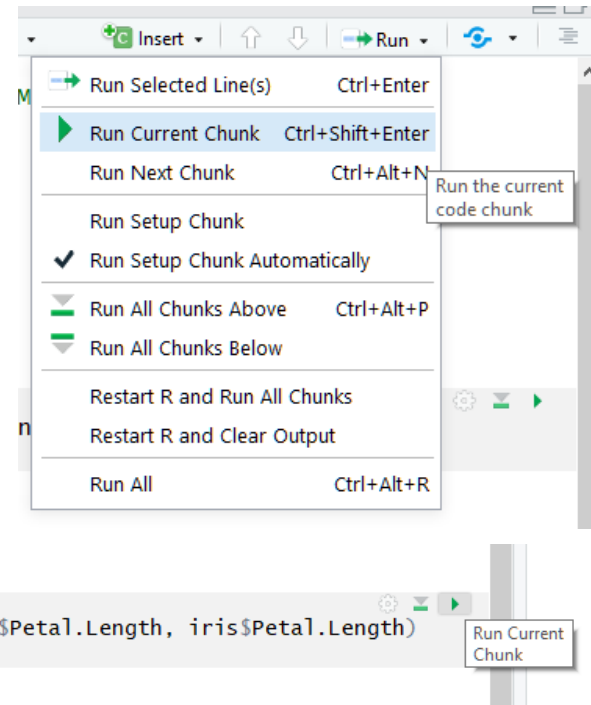
*Cmd+Option+i*



# Code

## Run a chunk (execute code)

- click "Run" and select an option from the list
- click the play symbol in the chunk corner
- place cursor within the chunk and press *Ctrl/Cmd+Shift+Enter*
- run one or more rows: Place the cursor in (or mark) the code you want to run and press *Ctrl/Cmd + Enter*

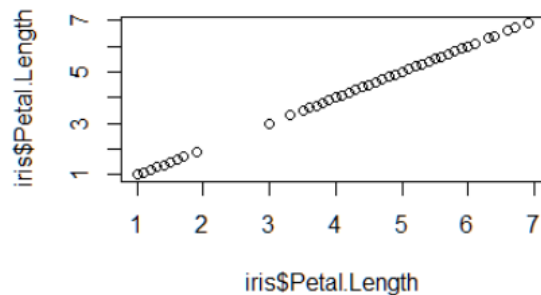


## [More keyboard shortcuts](#)

# Code

A chunk:

```
{r}  
plot(iris$Petal.Length, iris$Petal.Length)
```



This is plain text

type two spaces to make a line break

Simple formatting

- something in *\*italics\**
- something in **\*\*bold\*\***

Two ways of previewing the HTML document:

12:1

◆ (Top Level) ⇅

R Markdown ⇅

Console

*Developing skills in R*

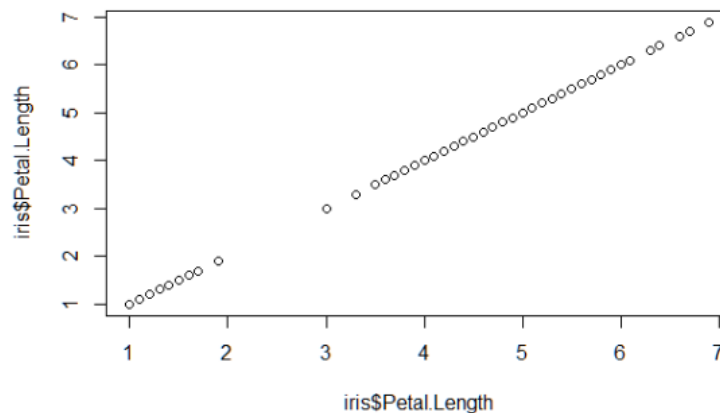
*Elsa Fogelström*

*13 November 2018*

A chunk:

Hide

```
plot(iris$Petal.Length, iris$Petal.Length)
```



This is plain text

type two spaces to make a line break

Simple formatting

- something in *italics*

## More formatting

# Knitr options

A chunk:

```
{r}
plot(iris$Petal.Length, iris$Petal.Length)
```

We are here

## Show/hide code and output

Hide code: `{r echo=FALSE}`

Hide code and output: `{r include=FALSE}`

Don't run chunk: `{r eval=FALSE}`



# Knitr options

## Figure size/position

Figure size: `{r fig.width = 3, fig.height = 4}`

Figure position: `{r fig.align = 'default'}`

Options: `'left', 'right', 'center'`

## supress messages and warnings

```
{r warning=FALSE,message=FALSE}
```

# Knitr options

## Set knitr options for the entire document

using `knitr::opts_chunk$set()` within an R code chunk:

```
knitr::opts_chunk$set(echo = FALSE, # hide code
                      # align figures left
                      # (does not work with word):
                      fig.align = "left")
```

# Make a table using the `kable` function

```
# library (knitr)
# specify a model:
mod <- lm(Petal.Length ~ Species, data =iris)

# present model output as table
knitr::kable(summary(mod)$coefficients, digits=3, format = 'html')
```

|                   | Estimate | Std. Error | t value | Pr(> t ) |
|-------------------|----------|------------|---------|----------|
| (Intercept)       | 1.462    | 0.061      | 24.023  | 0        |
| Speciesversicolor | 2.798    | 0.086      | 32.510  | 0        |
| Speciesvirginica  | 4.090    | 0.086      | 47.521  | 0        |

Works for .html and .pdf documents. Does not work for making .docx documents.

`format` needs to be set to `format = 'latex'` when knitting to pdf

# Make a table using the `pander` function

```
library(pander)
panderOptions('round', 3)
pander(mod)
```

I use it a lot for making tables for word documents. Works for .pdf and .html documents too. Did not work well in this .html presentation.

Examples and formatting options can be found [here](#) and [here](#)

# Combine text and code

## Inline code

**If not to be executed:** surrounded with single backticks (`)

Inline code

**code to be executed** is surrounded by a single backtick and starts with a small 'r'

There are ``r nrow(iris)`` entries in the iris dataset

There are 150 entries in the iris dataset

# Images

Within an r-chunk:

```
knitr::include_graphics(  
  'Iris_virginica.jpg')
```

In markdown (text):

```
![an image](Iris_virginica.jpg)
```

Both methods work with either path to local file or URL to an online file.

Here are some [Tips and tricks for working with images and figures in R Markdown documents](#)



# Embedded links

Add the description within brackets and the URL within parentheses:

```
[Iris versicolor] (http://www.florafinder.com/LargePhotos/D9/Iris_versicolor-81C439DB39.jpg)
```

It will end up looking something like this in the output document:

[Iris versicolor](http://www.florafinder.com/LargePhotos/D9/Iris_versicolor-81C439DB39.jpg)

# Rendering ('knitting') files

Render: the process of producing an output file from your .Rmd file

**knitr** executes code and saves your output to a markdown (.md) document

**pandoc** converts the markdown document to format of your choice

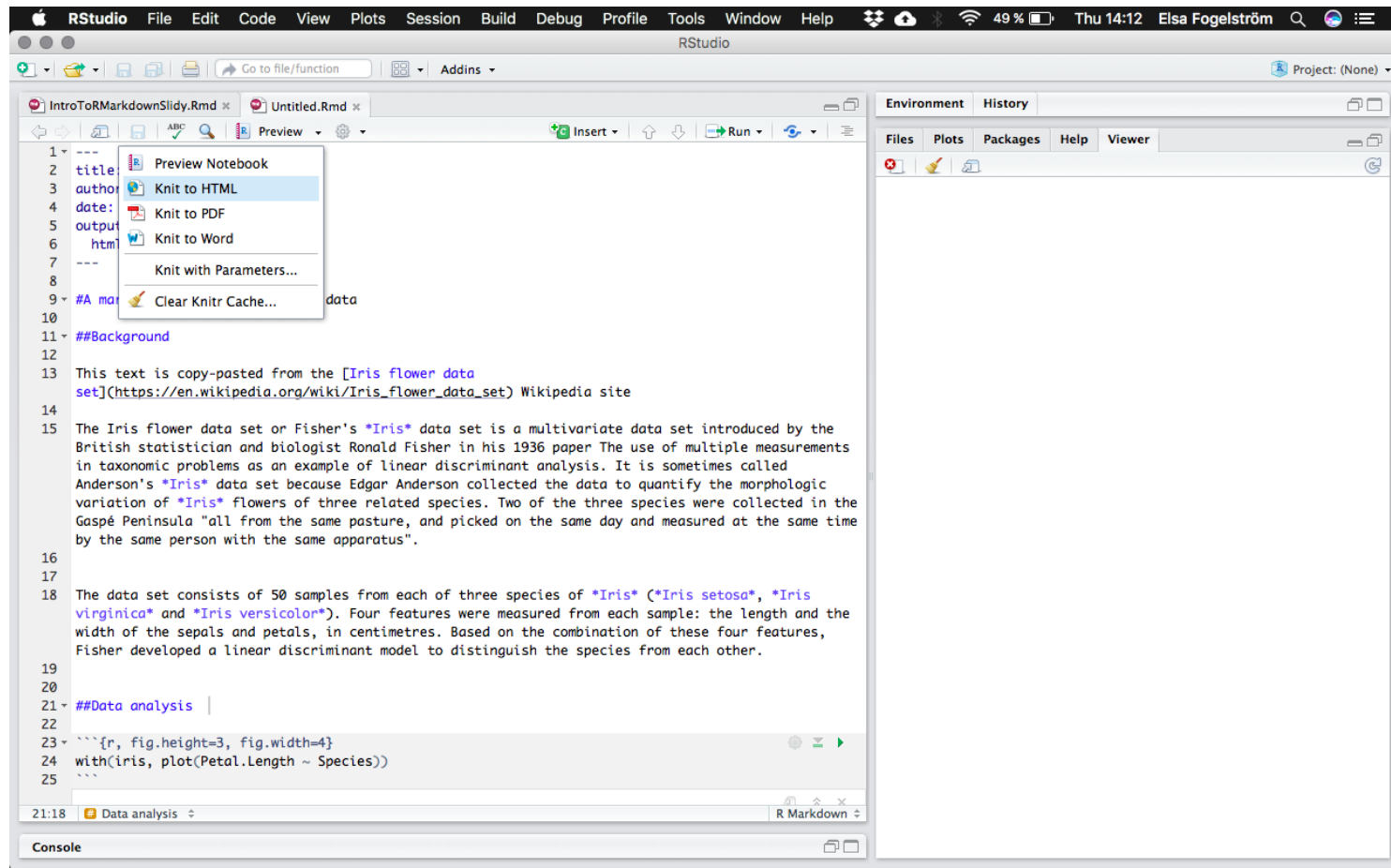


<https://d33wubrfki0l68.cloudfront.net/61d189fd9cdf955058415d3e1b28dd60e1bd7c9b/b739c/lesson-images/rmarkdownflow.png>

use `render` function or simply click *knit* (in drop-down menu next to *Preview* button)



# Render document: HTML



# Render document: HTML

**Intro to R Markdown**  
*Elsa Fogelström*  
17 November 2017

## A markdown example with *Iris* data

### Background

This text is copy-pasted from the [Iris flower data set](#) Wikipedia site

The Iris flower data set or Fisher's *Iris* data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis. It is sometimes called Anderson's *Iris* data set because Edgar Anderson collected the data to quantify the morphologic variation of *Iris* flowers of three related species. Two of the three species were collected in the Gaspé Peninsula "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus".

The data set consists of 50 samples from each of three species of *Iris* (*Iris setosa*, *Iris virginica* and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other.

### Data analysis

```
with(iris, plot(Petal.Length ~ Species))
```

Box plot showing Petal.Length (Y-axis, 3 to 7) versus Species (X-axis: setosa, virginica, versicolor). The plot displays the distribution of Petal.Length for each species, with individual data points overlaid on the box plots.

# Render document: PDF

RStudio View PDF

Page: 1 of 1 Automatic Zoom

## Intro to R Markdown

Elsa Fogelström  
17 November 2017

### A markdown example with *Iris* data

#### Background

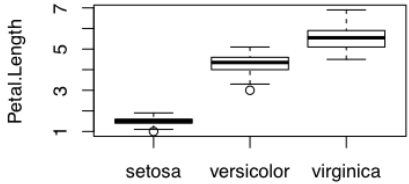
This text is copy-pasted from the *Iris* flower data set Wikipedia site

The *Iris* flower data set or Fisher's *Iris* data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis. It is sometimes called Anderson's *Iris* data set because Edgar Anderson collected the data to quantify the morphologic variation of *Iris* flowers of three related species. Two of the three species were collected in the Gaspé Peninsula "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus".

The data set consists of 50 samples from each of three species of *Iris* (*Iris setosa*, *Iris virginica* and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other.

#### Data analysis

```
with(iris, plot(Petal.Length ~ Species))
```



A box plot showing the distribution of Petal.Length for three species: setosa, versicolor, and virginica. The y-axis is labeled 'Petal.Length' and ranges from 1 to 7. The x-axis is labeled 'Species' and has categories 'setosa', 'versicolor', and 'virginica'. The plot shows that the setosa species has a much lower median Petal.Length (around 1.5) compared to the other two species, which have higher medians (around 4.5 for versicolor and 5.5 for virginica). The versicolor and virginica species show more overlap in their distributions.

| Species    | Median | Q1  | Q3  | Min | Max |
|------------|--------|-----|-----|-----|-----|
| setosa     | 1.5    | 1.2 | 1.8 | 1.0 | 2.2 |
| versicolor | 4.5    | 4.2 | 4.8 | 3.2 | 5.2 |
| virginica  | 5.5    | 5.2 | 5.8 | 4.8 | 6.2 |

# Render document: PDF

RStudio View PDF

Page: 1 of 1 Automatic Zoom

## Intro to R Markdown

Elsa Fogelström  
17 November 2017

### A markdown example with *Iris* data

#### Background

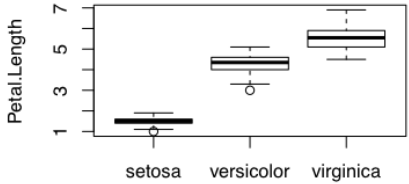
This text is copy-pasted from the *Iris* flower data set Wikipedia site

The *Iris* flower data set or Fisher's *Iris* data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis. It is sometimes called Anderson's *Iris* data set because Edgar Anderson collected the data to quantify the morphologic variation of *Iris* flowers of three related species. Two of the three species were collected in the Gaspé Peninsula "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus".

The data set consists of 50 samples from each of three species of *Iris* (*Iris setosa*, *Iris virginica* and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other.

#### Data analysis

```
with(iris, plot(Petal.Length ~ Species))
```



A box plot showing the distribution of Petal.Length for three species: setosa, versicolor, and virginica. The y-axis is labeled 'Petal.Length' and ranges from 1 to 7. The x-axis is labeled 'Species' and has categories 'setosa', 'versicolor', and 'virginica'. The plot shows that the setosa species has a much lower median Petal.Length (around 1.5) compared to the other two species, which have higher medians (around 4.5 for versicolor and 5.5 for virginica). The versicolor and virginica species show more overlap in their distributions.

| Species    | Min | Q1  | Median | Q3  | Max |
|------------|-----|-----|--------|-----|-----|
| setosa     | 1.0 | 1.4 | 1.5    | 1.6 | 1.8 |
| versicolor | 3.0 | 4.3 | 4.5    | 4.7 | 5.0 |
| virginica  | 4.5 | 5.1 | 5.5    | 5.7 | 6.0 |

# Render document: .docx

The screenshot shows the LibreOffice Writer application window titled "Untitled.docx". The interface includes a menu bar (File, Edit, View, Insert, Format, Styles, Table, Tools, Window, Help), a toolbar with various editing and formatting icons, and a sidebar on the right with icons for document structure and navigation. The main editing area displays the rendered content of a markdown file. The left sidebar shows the source markdown code, which includes a date, output paths, a title, author, date, and two sections: "Background" and "Data analysis". The rendered document in the center features a title "Intro to R Markdown" in a large blue font, followed by the author "Elsa Fogelström" and the date "17 November 2017". Below this is a section header "A markdown example with *Iris* data" in blue, followed by a sub-section "Background" in blue. The text under "Background" states: "This text is copy-pasted from the [Iris flower data set](#) Wikipedia site". The next paragraph describes the Iris flower data set, mentioning Ronald Fisher, 1936, and the Gaspé Peninsula. The bottom status bar indicates "Page 1 of 2", "229 words, 1,380 characters", "Default Style", "English (USA)", "Outline 1", and a zoom level of "90%".

LibreOffice File Edit View Insert Format Styles Table Tools Window Help

Untitled.docx

Heading 1 Calibri 16

Intro to R Markdown

Elsa Fogelström  
17 November 2017

A markdown example with *Iris* data

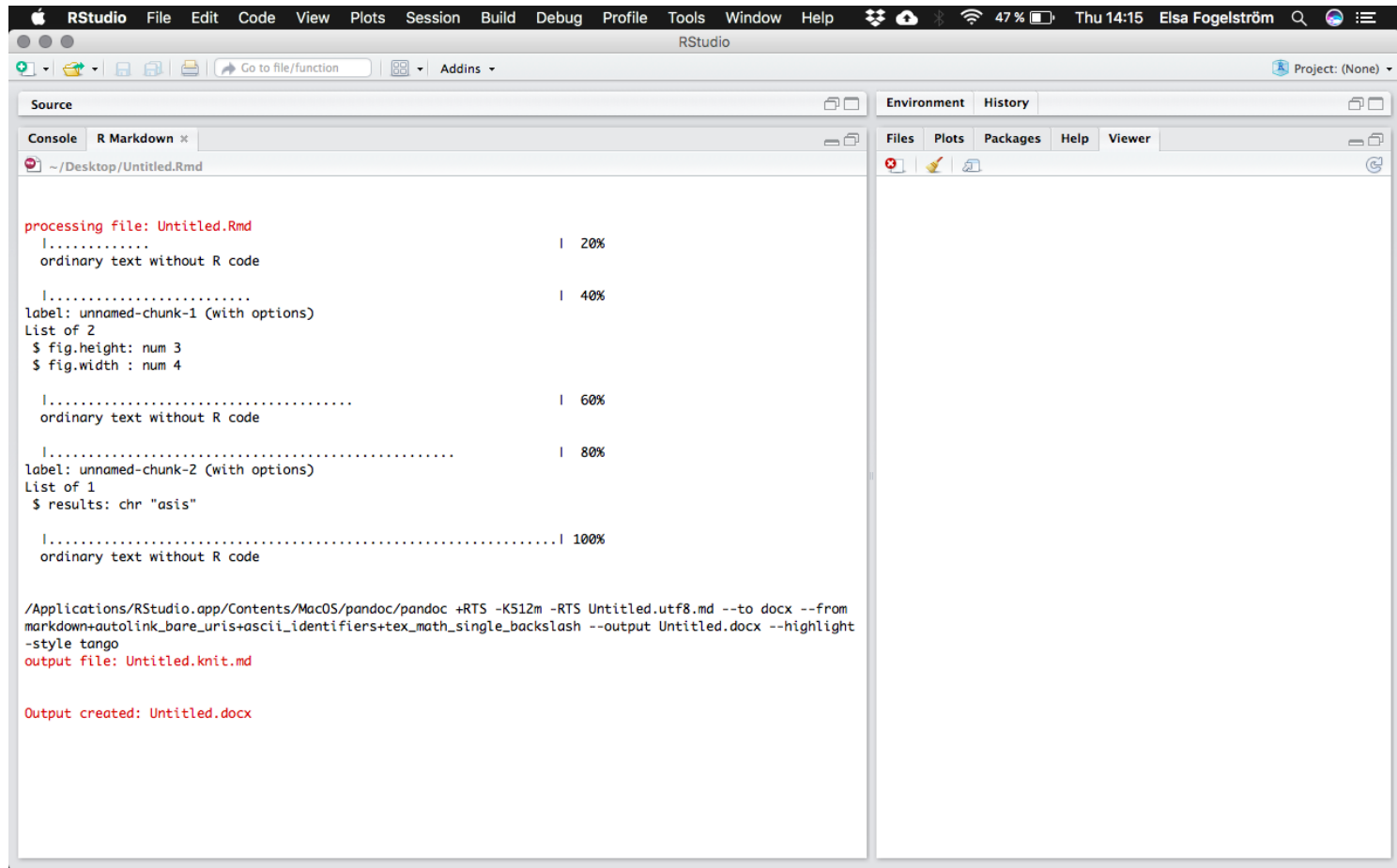
Background

This text is copy-pasted from the [Iris flower data set](#) Wikipedia site

The Iris flower data set or Fisher's *Iris* data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis. It is sometimes called Anderson's *Iris* data set because Edgar Anderson collected the data to quantify the morphologic variation of *Iris* flowers of three related species. Two of the three species were collected in the Gaspé Peninsula "all from the same pasture and picked on the same day and measured at the same time

Page 1 of 2 229 words, 1,380 characters Default Style English (USA) Outline 1 90%

# Render document



The screenshot shows the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, Help, and system status icons. The toolbar below the menu has icons for file operations and a 'Go to file/function' search bar. The main editor area is titled 'Source' and shows a file named 'Untitled.Rmd' at the path '~/Desktop/Untitled.Rmd'. The console output displays the progress of rendering the document to a .docx file. It shows two chunks of text being processed, with progress indicators at 20%, 40%, 60%, 80%, and 100%. The final output is 'Untitled.docx'. The right sidebar contains tabs for 'Environment', 'History', 'Files', 'Plots', 'Packages', 'Help', and 'Viewer', all of which are currently empty.

```
processing file: Untitled.Rmd
|.....| 20%
ordinary text without R code

|.....| 40%
label: unnamed-chunk-1 (with options)
List of 2
 $ fig.height: num 3
 $ fig.width : num 4

|.....| 60%
ordinary text without R code

|.....| 80%
label: unnamed-chunk-2 (with options)
List of 1
 $ results: chr "asis"

|.....| 100%
ordinary text without R code

/Applications/RStudio.app/Contents/MacOS/pandoc/pandoc +RTS -K512m -RTS Untitled.utf8.md --to docx --from
markdown+autolink_bare_uris+ascii_identifiers+tex_math_single_backslash --output Untitled.docx --highlight
-style tango
output file: Untitled.knit.md

Output created: Untitled.docx
```

# Try it for yourselves!

1. Create a new .Rmd file
2. Play around with as much formatting as possible:
  - edit the YAML metadata
  - headers
  - **bold**, *italics*, lists
  - inline code
  - figures from data (using your own data or the `iris` dataset)
  - mathematical annotation
  - visible/hidden r code
  - ...
3. Click the *Preview* button to see how it turned out
4. Render your report to different document formats

# If you run into problems

## Is everything in the right place?

Within/outside code chunks?

## Error message:

Search online! You're probably not the first to get that error message

## Best practices and solutions to common problems:

<https://rmd4sci.njtierney.com/common-problems-with-rmarkdown-and-some-solutions.html>



# Problems with LaTeX and knitting to .pdf

## TinyTex

Error message when installing on mac: <https://github.com/yihui/tinytex/issues/24>

Update or reinstall TinyTex <https://yihui.name/tinytex/r/#debugging>

If TinyTex doesn't work, try using [texworks](#)

## Knitting to pdf: Error messages

- Try to figure out problem from message
  - code chunk that doesn't run?
  - weird symbols?
- File/Save with encoding/UTF-8
- Change LaTeX engine to 'xelatex' (output options/advanced)
- Ask Google

- This is an R Markdown presentation.
- Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.
- This presentation was created in RStudio using [remarkjs](#) framework through R package [xaringan](#) .
- This template uses custom CSS style.
- For R Markdown, see <http://rmarkdown.rstudio.com>
- For R Markdown presentations, see <https://rmarkdown.rstudio.com/lesson-11.html>

# References/tutorials

[R Markdown cheat sheet](#)

[Introduction to R Markdown \(RStudio\)](#)

[R Markdown: The Definitive Guide \(Yihui Xie, J. J. Allaire, Garrett Golemund\)](#)

[Knitr output options \(Yihui Xie\)](#)

# Try it for yourselves!

1. Create a new .Rmd file
2. Play around with as much formatting as possible:
  - edit the YAML metadata
  - headers
  - **bold**, *italics*, lists
  - inline code
  - figures from data (using your own data or the `iris` dataset)
  - mathematical annotation
  - visible/hidden r code
  - ...
3. Click the *Preview* button to see how it turned out
4. Render your report to different document formats

**Thank you**