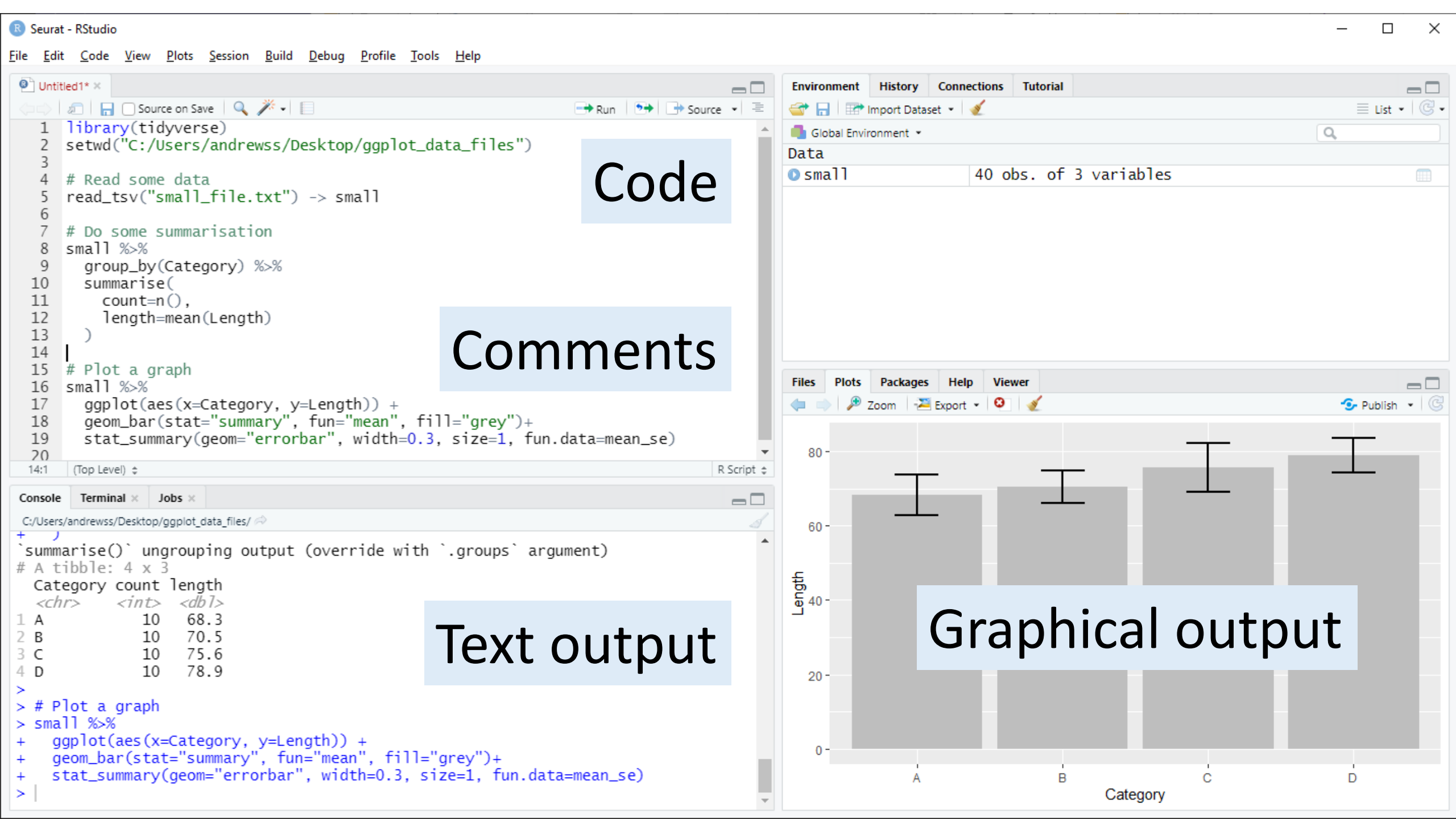


# Using R Notebooks

Simon Andrews

v2022-03





# Problems with conventional scripts

- Only the code is generally distributed
  - Output not included – users have to run it again
- No collation of output
  - Can't see which bit of code generated what output
  - No automated saving of results
- Limited commenting
  - Text comments, no formatting or structure

# R Notebooks

- Alternative document format to conventional scripts
- Collates into a single document
  - Code
  - Formatted commentary
  - Output (text and graphical)
- Exported to HTML, PDF or Word

# Code

Introduction

Processing

Read the data

Summarise

Plot

```
1 ---
2 title: "Example Notebook"
3 output:
4   html_document:
5     df_print: paged
6     toc: true
7     toc_float: true
```

10 Introduction

11 =====

13 This is an example of a notebook to show how they work.

```
15 ```{r message=FALSE}
16 library(tidyverse)
17 ```
```

19 Processing

20 =====

22 Read the data

23 -----

```
25 ```{r message=FALSE}
26 read_tsv("small_file.txt") -> small
27 head(small)
28 ```
```

Sample <chr>	Length <dbl>	Category <chr>
-----------------	-----------------	-------------------

x_1	45	A
-----	----	---

x_2	82	B
-----	----	---

x_3	81	C
-----	----	---

x_4	56	D
-----	----	---

x_5	96	A
-----	----	---

## Summarise

We're going to calculate the mean of the lengths per category

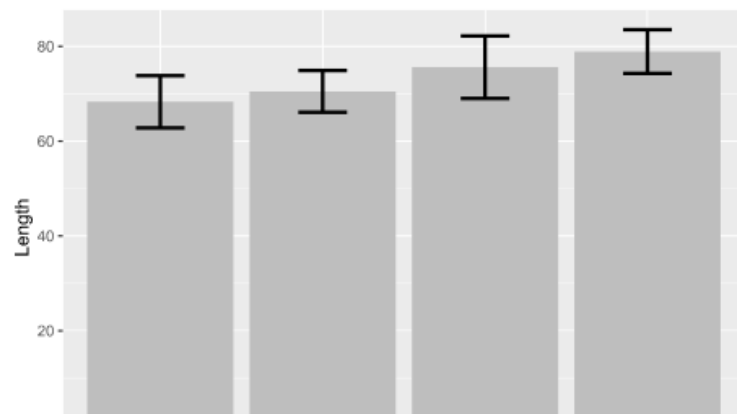
```
small %>%
  group_by(Category) %>%
  summarise(
    count=n(),
    length=mean(Length)
  )
```

Category <chr>	count <int>	length <dbl>
A	10	68.3
B	10	70.5
C	10	75.6
D	10	78.9

4 rows

## Plot

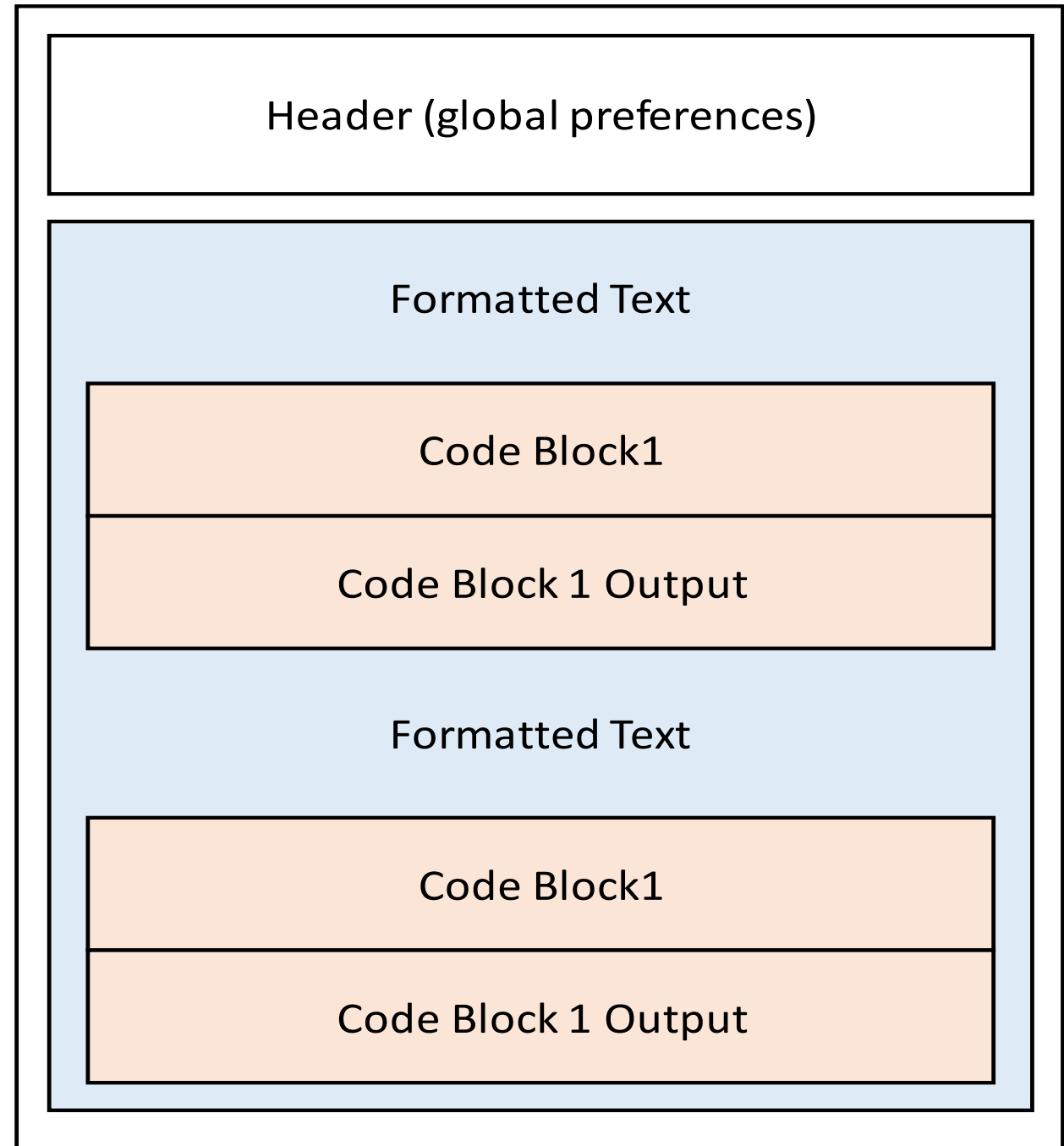
```
small %>%
  ggplot(aes(x=Category, y=Length)) +
  geom_bar(stat="summary", fun="mean", fill="grey") +
  stat_summary(geom="errorbar", width=0.3, size=1, fun.data=mean_se)
```



# Output

# Notebook Structure

- Single overall text document, split into sections
  - Header (mostly preferences)
  - Body
    - Commentary (default)
    - R Code
    - Output (graphical and text)





# Notebook sections

Header

Commentary

Code

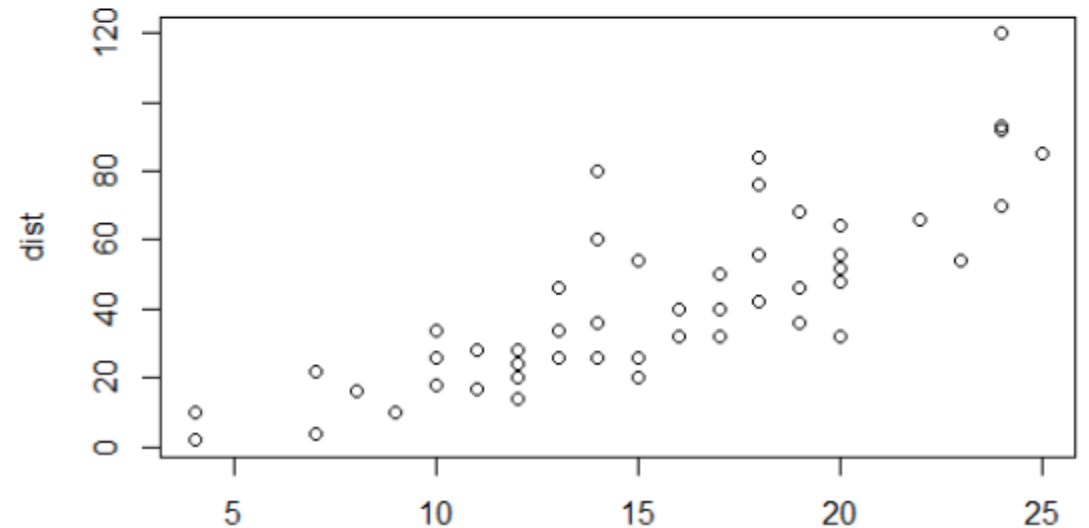
```
1 ---
2 title: "R Notebook"
3 output: html_notebook
4 ---
5
6 This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook.
7 When you execute code within the notebook, the results appear
8 beneath the code.
9
10 Try executing this chunk by clicking the *Run* button within the
11 chunk or by placing your cursor inside it and pressing
12 *Ctrl+Shift+Enter*.
```

Sections are marked by special quotes

---  
--- for header

```\{r}\`\`\`  
```\{r}\`\`\` for R code

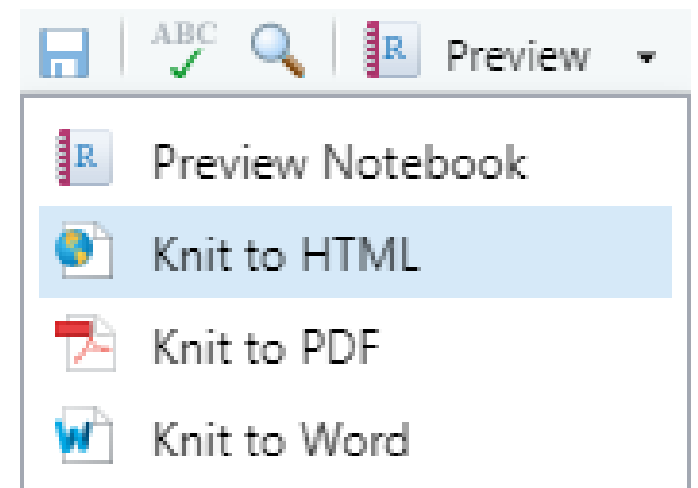
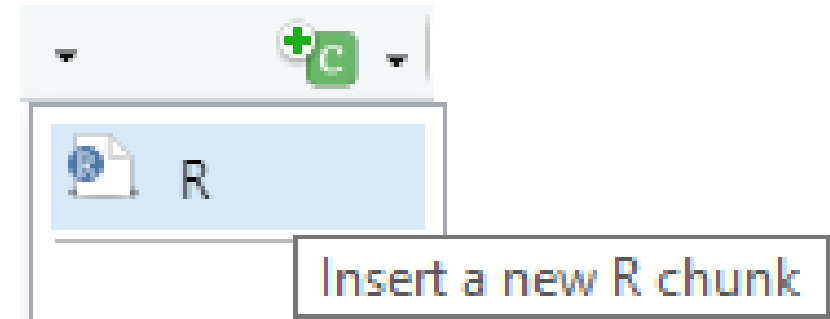
Default for unquoted text is commentary





# Notebook workflow

- Create new notebook document
- Save it straight away (use a .Rmd extension)
- Add commentary in Markdown format
- Add R sections using Insert > R
- Run code blocks to generate output
  
- Knit document to HTML / PDF / Word



Be careful not to delete any of the section markers added by 'insert' or the header

# Running R code in a notebook

- Control + Return runs one line
  - Output goes below
  - Output replaces any previous block output
- Control + Shift + Return runs the block
  - Multiple outputs put into clickable windows
  - Will be interspersed in compiled document
  - Can also press the 'play' button at top right

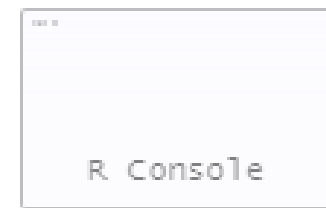
```
```{r}
tibble(x=1:5) -> some.data

some.data

some.data %>% pull(x) %>% mean()
```
```



tbl\_df  
5 x 1



| x<br><int> |
|------------|
| 1          |
| 2          |
| 3          |
| 4          |
| 5          |

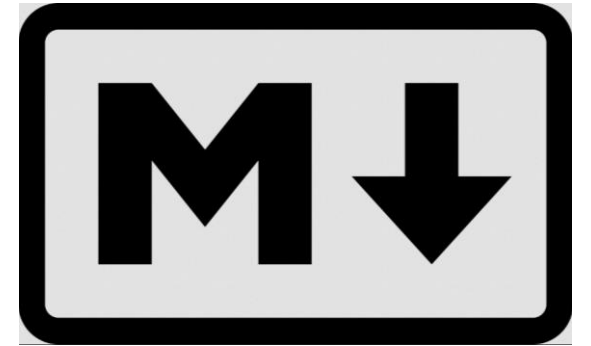
5 rows

# Exercise 1

# Using Markdown

# Commentary sections use 'Markdown'

- Simple markup language
- Designed to be nicely readable as plain text
- Compiles to properly formatted text
- Simple syntax



# Markdown basics

- Headings

```
# Heading1  
## Heading 2  
### Heading 3 etc.
```

```
Heading 1
```

```
=====
```

```
Heading 2
```

```
-----
```

- Lists (need a blank line first)

```
* Bullet 1
```

```
[Tab] * Sub-bullet 1
```

```
* Bullet 2
```

```
1. Numbered 1
```

```
2. Numbered 2
```



Headings also give you navigation for your document, so they're worth using!

# Markdown basics

- Emphasis

*\*italics\**  
*\_italics\_*

**\*\*bold\*\***  
**\_\_bold\_\_**

***\*\*\*bold italics\*\*\****  
***\_\_\_bold italics\_\_\_***

`vol=width\*depth\*height`

NOT bold (escaped)

- Other formatting

````fixed width code etc````

`> quoted text`

`super^script^`

`sub~script~`

`*****` or `-----` page break

Needs blank line above and below

# Markdown basics

- Tables

| Name   | Quest                      | Success   |  |
|--------|----------------------------|-----------|--|
| :----- | :-----:                    | -----:    |  |
| Simon  | To teach R                 | Sometimes |  |
| Emma   | To teach the world to sing | Always    |  |
| Libby  | To pass her GCSEs          | Unknown   |  |

:--- Left Justified

:--: Centred

---: Right Justified



# Markdown basics

$$e = mc^2$$

`$e=mc^2$`

$$\sum_{i=1}^n X_i$$

`$$\sum_{i=1}^n X_i$`

$$F_{i,j}$$

`$F_{i,j}$`

$$\sqrt{x^2 - 5y}$$

`$$\sqrt{x^2 - 5y}$`

$$\sum_{i=1}^n \left( \frac{X_i}{Y_i} \right)$$

`$$\sum_{i=1}^n \left( \frac{X_i}{Y_i} \right)$`

- Markdown supports Latex equations.
  - `$equation$` is inline with text
  - `$$equation$$` is as a separate block

# Exercise 2

R code block details

# Working directories

- Working directories
  - Working directory is automatically set to directory with Rmd file
  - That's why we immediately save
  - Designed so that data and code all go together
  - Can run `setwd` but get a warning, and only lasts for 1 block

# Good code block practices

- Break code into short chunks
- All chunks are part of the same session
- Stop the block as soon as any output is generated

```
```{r}
tibble(x=1:5) -> some.data
some.data
```
```

| x     |
|-------|
| <int> |
| 1     |
| 2     |
| 3     |
| 4     |
| 5     |

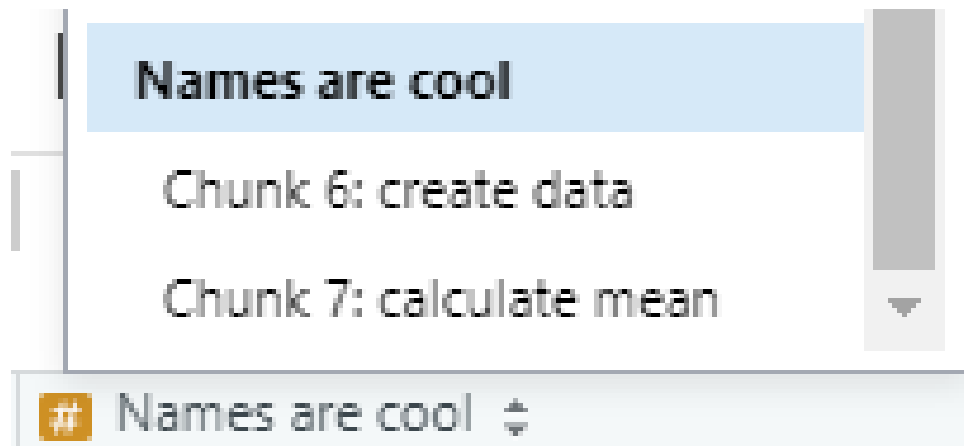
5 rows

```
```{r}
some.data %>% pull(x) %>% mean()
|```
```

[1] 3

# Good code block practices

- Name your chunks
- Name appears in the navigation along with headings you've created



```
Names are cool
```

```
-----
```

```
```{r "create data"}
```

```
tibble(x=1:5) -> some.data
```

```
some.data
```

```
```
```

```
```{r "calculate mean"}
```


```
some.data %>% pull(x) %>% mean()
```

```
```
```

# Displaying tibbles

- By default you don't see the text form of tibbles/dataframes
- You get a nice interactive table
  - Not in all output formats
- Buttons to see more columns/rows

```
```{r}
read_csv("Child_Variants.csv") -> child
child
```
```



The screenshot shows two windows from an R environment. The 'R Console' window on the left contains the R code for reading a CSV file. The 'spec\_tbl\_df' window on the right shows the structure of the resulting tibble: 'spec\_tbl\_df' with dimensions '25822 x 11'.

| CHR<br><dbl> | POS<br><dbl> | dbSNP<br><chr> | REF<br><chr> | ALT<br><chr> | QUAL<br><dbl> | GENE<br><chr> |
|--------------|--------------|----------------|--------------|--------------|---------------|---------------|
| 1            | 69270        | .              | A            | G            | 16            | OR4F5         |
| 1            | 69511        | rs75062661     | A            | G            | 200           | OR4F5         |
| 1            | 69761        | .              | A            | T            | 200           | OR4F5         |
| 1            | 69897        | rs75758884     | T            | C            | 59            | OR4F5         |
| 1            | 877831       | rs6672356      | T            | C            | 200           | SAMD11        |
| 1            | 881627       | rs2272757      | G            | A            | 200           | NOC2L         |
| 1            | 887801       | rs3828047      | A            | G            | 200           | NOC2L         |
| 1            | 888639       | rs3748596      | T            | C            | 200           | NOC2L         |
| 1            | 888659       | rs3748597      | T            | C            | 200           | NOC2L         |
| 1            | 889158       | rs13303056     | G            | C            | 200           | NOC2L         |

1-10 of 25,822 rows | 1-7 of... Previous  2 3 4 5 6 ... 10

# Displaying tibbles

- Although you only see 10 rows, all of the data goes into your document
- When rendered to HTML / PDF this can make your document BIG
- Use the `head()` function to only show a few example rows

```
read_csv("Child_Variants.csv") -> child
head(child, n=20)
```

| CHR<br><dbl> | POS<br><dbl> | dbSNP<br><chr> | REF<br><chr> | ALT<br><chr> | QUAL<br><dbl> | GENE<br><chr> |
|--------------|--------------|----------------|--------------|--------------|---------------|---------------|
| 1            | 69270        | .              | A            | G            | 16            | OR4F5         |
| 1            | 69511        | rs75062661     | A            | G            | 200           | OR4F5         |
| 1            | 69761        | .              | A            | T            | 200           | OR4F5         |
| 1            | 69897        | rs75758884     | T            | C            | 59            | OR4F5         |
| 1            | 877831       | rs6672356      | T            | C            | 200           | SAMD11        |
| 1            | 881627       | rs2272757      | G            | A            | 200           | NOC2L         |
| 1            | 887801       | rs3828047      | A            | G            | 200           | NOC2L         |
| 1            | 888639       | rs3748596      | T            | C            | 200           | NOC2L         |
| 1            | 888659       | rs3748597      | T            | C            | 200           | NOC2L         |
| 1            | 889158       | rs13303056     | G            | C            | 200           | NOC2L         |

1-10 of 20 rows | 1-7 of 11 columns

Previous  2 Next



# Controlling warnings / errors / messages

```
{r}  
library(tidyverse)
```

Registered S3 methods overwritten by 'dbplyr':

```
method      from  
print.tbl_lazy  
print.tbl_sql
```

-- Attaching packages ----- tidyverse 1.3.0 --

```
✓ ggplot2 3.3.2      ✓ purrr   0.3.4  
✓ tibble  3.0.4      ✓ dplyr   1.0.2  
✓ tidyr   1.1.2      ✓ stringr 1.4.0  
✓ readr   1.4.0      ✓ forcats 0.5.0
```

-- Conflicts ----- tidyverse\_conflicts() --

```
x dplyr::filter() masks stats::filter()  
x dplyr::lag()    masks stats::lag()
```

```
{r}  
read_tsv("small_file.txt") -> small
```

```
-- Column specification -----  
cols(  
  Sample = col_character(),  
  Length = col_double(),  
  Category = col_character()  
)
```

# Controlling warnings / errors / messages

- Can select which output you want to see using the block header

```
```{r "Block name", warnings=FALSE}
```

- Can remove

- Warnings `{r warnings=FALSE}`
- Errors `{r error=TRUE}` means that script doesn't stop on error
- Messages `{r message=FALSE}`
- Code `{r echo=FALSE}`
- Code + output `{r include=FALSE}`

# Changing graphics options

- You can change the way that figures / graphs are displayed by changing R code block options

- Change the file format (default is PNG)

```
```{r dev="svg"}
```

- Change the size

```
```{r fig.height=5, fig.width=8}
```

- Change the alignment (only affected compiled document)

```
```{r fig.align="center"}
```

- Add a legend

```
```{r fig.cap="This is a great picture"}
```

# Exercise 3

# Changing document appearance

# Table of Contents

- If you have used headings in your document then you can auto-create a table of contents
- This can be a fixed set of links at the top of your document, or a floating table on the left
- This is set in the header section

Introduction
Processing
<b>Read the data</b>
Summarise
Plot
Names are cool

## Processing

### Read the data

```
read_tsv("small_file.txt")
```

```
##  
## -- Column specification  
## cols(
```

```
---  
title: "Example Notebook"  
output:  
  html_document:  
    df_print: paged  
    toc: yes  
    toc_float: yes  
---
```

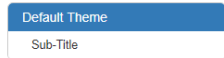
# Document themes

- HTML documents are based on the bootswatch theme collection (<https://bootswatch.com>)
- You can change the theme by adding to the header

```
---  
title: "Themes"  
output:  
  html_document:  
    df_print: paged  
    toc: true  
    toc_float: true  
    theme: yeti  
    highlight: kate
```

```
---
```

# Document themes



## Themes

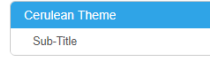
### Default Theme

This is a small document to show the effect of changing the themes.

#### Sub-Title

- Themes can change
- The overall appearance

of your document in a quick and easy fashion.



## Themes

### Cerulean Theme

This is a small document to show the effect of changing the themes.

#### Sub-Title

- Themes can change
- The overall appearance

of your document in a quick and easy fashion.



## Themes

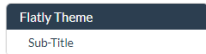
### Journal Theme

This is a small document to show the effect of changing the themes.

#### Sub-Title

- Themes can change
- The overall appearance

of your document in a quick and easy fashion.



## Themes

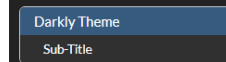
### Flatly Theme

This is a small document to show the effect of changing the themes.

#### Sub-Title

- Themes can change
- The overall appearance

of your document in a quick and easy fashion.



## Themes

### Darkly Theme

This is a small document to show the effect of changing the themes.

#### Sub-Title

- Themes can change
- The overall appearance

of your document in a quick and easy fashion.



## Themes

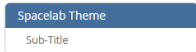
### Readable Theme

This is a small document to show the effect of changing the themes.

#### Sub-Title

- Themes can change
- The overall appearance

of your document in a quick and easy fashion.



## Themes

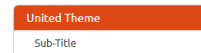
### Spacelab Theme

This is a small document to show the effect of changing the themes.

#### Sub-Title

- Themes can change
- The overall appearance

of your document in a quick and easy fashion.



## Themes

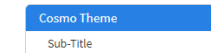
### United Theme

This is a small document to show the effect of changing the themes.

#### Sub-Title

- Themes can change
- The overall appearance

of your document in a quick and easy fashion.



## Themes

### Cosmo Theme

This is a small document to show the effect of changing the themes.

#### Sub-Title

- Themes can change
- The overall appearance

of your document in a quick and easy fashion.

(there are more than this)



# Highlighting themes

- Similarly to the document themes you can also change the colouring / style used to highlight R code in your document

```
---  
title: "Themes"  
output:  
  html_document:  
    df_print: paged  
    toc: true  
    toc_float: true  
    theme: yeti  
    highlight: kate  
---
```

# Highlighting themes

```
# Haddock  
  
library(tidyverse)  
  
starwars %>%  
  filter(height>150) %>%  
  arrange(desc(birth_year)) %>%  
  filter(gender=="male")
```

```
# Zenburn  
  
library(tidyverse)  
  
starwars %>%  
  filter(height>150) %>%  
  arrange(desc(birth_year)) %>%  
  filter(gender=="male")
```

```
# Espresso  
  
library(tidyverse)  
  
starwars %>%  
  filter(height>150) %>%  
  arrange(desc(birth_year)) %>%  
  filter(gender=="male")
```

```
# Monochrome  
  
library(tidyverse)  
  
starwars %>%  
  filter(height>150) %>%  
  arrange(desc(birth_year)) %>%  
  filter(gender=="male")
```

```
# Kate  
  
library(tidyverse)  
  
starwars %>%  
  filter(height>150) %>%  
  arrange(desc(birth_year)) %>%  
  filter(gender=="male")
```

```
# Textmate  
  
library(tidyverse)  
  
starwars %>%  
  filter(height>150) %>%  
  arrange(desc(birth_year)) %>%  
  filter(gender=="male")
```

```
# Tango  
  
library(tidyverse)  
  
starwars %>%  
  filter(height>150) %>%  
  arrange(desc(birth_year)) %>%  
  filter(gender=="male")
```

```
# Pygments  
  
library(tidyverse)  
  
starwars %>%  
  filter(height>150) %>%  
  arrange(desc(birth_year)) %>%  
  filter(gender=="male")
```

# Tibble / DataFrame display options

- Rather than text output you see an interactive HTML version of tibbles
  - This will vary by output document type
- A few options exist for how they are displayed these are set in the header, and are specific to the HTML output type:

```
html_document:  
  df_print: paged
```

# Tibble / DataFrame display options

Only works on  
data frames

<b>Option value</b>	<b>Text or HTML</b>	<b>Fits to space</b>	<b>Restricts rows</b>	<b>Paging controls</b>
default	text	no	no	no
kable	HTML	no	no	no
tibble	text	yes	yes	no
paged	HTML	yes	yes	yes

This is the default

# Tibble / DataFrame display options

## Tibble

```
## # A tibble: 20 x 5
##   Plant Type   Treatment   conc uptake
##   <chr> <chr> <chr>      <dbl> <dbl>
## 1 Qn1   Quebec nonchilled    95    16
## 2 Qn1   Quebec nonchilled   175   30.4
## 3 Qn1   Quebec nonchilled   250   34.8
## 4 Qn1   Quebec nonchilled   350   37.2
## 5 Qn1   Quebec nonchilled   500   35.3
## 6 Qn1   Quebec nonchilled   675   39.2
## 7 Qn1   Quebec nonchilled  1000   39.7
## 8 Qn2   Quebec nonchilled    95   13.6
## 9 Qn2   Quebec nonchilled   175   27.3
## 10 Qn2   Quebec nonchilled   250   37.1
## 11 Qn2   Quebec nonchilled   350   41.8
## 12 Qn2   Quebec nonchilled   500   40.6
## 13 Qn2   Quebec nonchilled   675   41.4
## 14 Qn2   Quebec nonchilled  1000   44.3
## 15 Qn3   Quebec nonchilled    95   16.2
## 16 Qn3   Quebec nonchilled   175   32.4
## 17 Qn3   Quebec nonchilled   250   40.3
## 18 Qn3   Quebec nonchilled   350   42.1
## 19 Qn3   Quebec nonchilled   500   42.9
## 20 Qn3   Quebec nonchilled   675   43.9
```

## Kable

Plant	Type	Treatment	conc	uptake
Qn1	Quebec	nonchilled	95	16.0
Qn1	Quebec	nonchilled	175	30.4
Qn1	Quebec	nonchilled	250	34.8
Qn1	Quebec	nonchilled	350	37.2
Qn1	Quebec	nonchilled	500	35.3
Qn1	Quebec	nonchilled	675	39.2
Qn1	Quebec	nonchilled	1000	39.7
Qn2	Quebec	nonchilled	95	13.6
Qn2	Quebec	nonchilled	175	27.3
Qn2	Quebec	nonchilled	250	37.1
Qn2	Quebec	nonchilled	350	41.8
Qn2	Quebec	nonchilled	500	40.6
Qn2	Quebec	nonchilled	675	41.4
Qn2	Quebec	nonchilled	1000	44.3
Qn3	Quebec	nonchilled	95	16.2
Qn3	Quebec	nonchilled	175	32.4
Qn3	Quebec	nonchilled	250	40.3
Qn3	Quebec	nonchilled	350	42.1
Qn3	Quebec	nonchilled	500	42.9
Qn3	Quebec	nonchilled	675	43.9

## Paged

Plant	Type	Treatment	conc	uptake
<chr>	<chr>	<chr>	<dbl>	<dbl>
Qn1	Quebec	nonchilled	95	16.0
Qn1	Quebec	nonchilled	175	30.4
Qn1	Quebec	nonchilled	250	34.8
Qn1	Quebec	nonchilled	350	37.2
Qn1	Quebec	nonchilled	500	35.3
Qn1	Quebec	nonchilled	675	39.2
Qn1	Quebec	nonchilled	1000	39.7
Qn2	Quebec	nonchilled	95	13.6
Qn2	Quebec	nonchilled	175	27.3
Qn2	Quebec	nonchilled	250	37.1
Qn2	Quebec	nonchilled	350	41.8
Qn2	Quebec	nonchilled	500	40.6
Qn2	Quebec	nonchilled	675	41.4
Qn2	Quebec	nonchilled	1000	44.3
Qn3	Quebec	nonchilled	95	16.2
Qn3	Quebec	nonchilled	175	32.4
Qn3	Quebec	nonchilled	250	40.3
Qn3	Quebec	nonchilled	350	42.1
Qn3	Quebec	nonchilled	500	42.9
Qn3	Quebec	nonchilled	675	43.9

1-10 of 20 rows

Previous 1 2 Next

# Automating Notebook Rendering

# Generating a notebook programmatically

```
Rscript -e "rmarkdown::render('example.Rmd')"
```

# Adding notebook parameters

```
---  
title: My Document  
output: html_document  
params:  
  year: 2018  
  region: Europe  
  printcode: TRUE  
  data: "file.csv"  
---
```

Parameters are collected  
in a list called `params`

```
print(params$year)
```

```
[1] 2018
```



# Parameters can be R code

```
---  
title: My Document  
output: html_document  
params:  
  date: !r Sys.Date()  
  today: !r lubridate::today()  
---
```

You can use code from packages but need to supply the full function name, including package name

# Parameters can be supplied at runtime

```
---  
title: My Document  
output: html_document  
params:  
  year: 2018  
  printcode: TRUE  
  data: "file.csv"  
---
```

```
Rscript -e "rmarkdown::render(  
  'example.Rmd',  
  params=list(data='data.csv')  
)"
```

```
read_csv(params$data)
```

# Parameters can also be used in Markdown

```
---  
output:  
  html_document:  
    df_print: paged  
  
params:  
  file: "test.csv"  
  date: !r Sys.Date()  
---  
  
---  
title: `r params$date`  
---  
  
```${r results='asis', echo=FALSE}  
cat("# Processing file ",params$file)  
```
```

```
Rscript -e "rmarkdown::render(  
  'example.Rmd',  
  params=list(data="data.csv")  
)"
```

2020-11-18

Processing file data.csv

# Exercise 4